

Res'd PCT/PTO

RECD 2004

10/508884

PCT/JP03/01094

04.02.03

日 本 国 特 許 庁

JAPAN PATENT OFFICE

RECEIVED

21 FEB 2003

WIPO

PCT

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date Application:

2002年 6月21日

出 願 番 号

Application Number:

特願2002-180926

[ST.10/C]:

[JP2002-180926]

出 願 人

Applicant(s):

ソニー株式会社

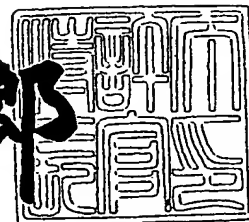
**PRIORITY  
DOCUMENT**

SUBMITTED OR TRANSMITTED IN  
COMPLIANCE WITH RULE 17.1(a) OR (b)

2002年11月29日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

太田信一郎



出証番号 出証特2002-3094556

Best Available Copy

【書類名】 特許願

【整理番号】 0290219217

【提出日】 平成14年 6月21日

【あて先】 特許庁長官殿

【国際特許分類】 H04L 29/00

【発明者】

    【住所又は居所】 東京都品川区北品川6丁目7番35号 ソニー株式会社  
    内

    【氏名】 浅井 淳

【特許出願人】

    【識別番号】 000002185

    【氏名又は名称】 ソニー株式会社

【代理人】

    【識別番号】 100082131

    【弁理士】

    【氏名又は名称】 稲本 義雄

    【電話番号】 03-3369-6479

【先の出願に基づく優先権主張】

    【出願番号】 特願2002- 84764

    【出願日】 平成14年 3月26日

【手数料の表示】

    【予納台帳番号】 032089

    【納付金額】 21,000円

【提出物件の目録】

    【物件名】 明細書 1

    【物件名】 図面 1

    【物件名】 要約書 1

    【包括委任状番号】 9708842

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 プログラム、記録媒体、情報処理装置および方法、並びに、情報処理システム

【特許請求の範囲】

【請求項1】 コンピュータに実行させるプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェアと、

前記ブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群、および、前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された前記マルチメディアコンテンツと

を含むことを特徴とするプログラム。

【請求項2】 前記ブラウザソフトウェアは、さらに、複数のメディアを同期再生するサービスと、

第 1 の前記シーンと、第 2 の前記シーンとの間を遷移する遷移サービスと  
 を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、  
 前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供する  
 ことを特徴とする請求項 1 に記載のプログラム。

【請求項 3】 前記マルチメディアコンテンツは、前記ブラウザソフトウェア  
 ア上で再生される場合、提示の一意性が保たれるように構成される  
 ことを特徴とする請求項 1 に記載のプログラム。

【請求項 4】 コンピュータに実行させるプログラムであって、  
 所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテン  
 ツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリ  
 プトのそれぞれをロードするサービスと、  
 前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理  
 するサービスと、

前記各コンポーネントのそれぞれを、所定の 3D 仮想空間に表示させ、かつ、  
 前記 3D 仮想空間に既に表示されている前記コンポーネントを消去させるサービ  
 スと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間  
 の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの  
 間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサー  
 ビスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、  
 前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウ  
 ザソフトウェアと、

前記ブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタ  
 マイズして任意に組み合わせた前記コンポーネント群、および、前記コンポーネ  
 ントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された  
 前記マルチメディアコンテンツと



を含むことを特徴とするコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項5】 コンピュータに実行させるマルチメディアコンテンツとしてのプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトと

を含む前記シーンが、複数個集合された

ことを特徴するプログラム。

【請求項6】 コンピュータに実行させるマルチメディアコンテンツとしてのプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スク

リプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトと

を含む前記シーンが、複数個集合された

ことを特徴するコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項7】 コンピュータに実行させるブラウザソフトウェアとしてのプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービ

スト、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供することを特徴とするプログラム。

【請求項 8】 コンピュータに実行させるブラウザソフトウェアとしてのプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供する

ことを特徴とするコンピュータが読み取り可能なプログラムが記録されている

記録媒体。

【請求項 9】 コンピュータに実行させるプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された前記マルチメディアコンテンツを製作する製作ステップを含むことを特徴とするプログラム。

【請求項 10】 製作された前記マルチメディアコンテンツと、前記ブラウザソフトウェアとを組み合わせ、1つのプログラムとする組み合わせステップをさらに含む

ことを特徴とする請求項 9 に記載のプログラム。

【請求項 11】 前記製作ステップは、

配布形式の前記コンポーネントを読み込んでGUI上に表示させるように制御す

る表示制御ステップと、

前記GUI上で選択された前記コンポーネントを前記マルチメディアコンテンツに追加する追加ステップと、

前記マルチメディアコンテンツ上に含まれる前記コンポーネントのカスタマイズを行う前記GUIを提供する提供ステップと、

前記表示制御ステップ、前記追加ステップ、および、前記提供ステップの処理により編集された前記マルチメディアコンテンツを、配布可能な形式に変換する変換ステップと

を含むことを特徴とする請求項9に記載のプログラム。

【請求項12】 コンピュータに実行させるプログラムであって、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、

複数個集合された前記マルチメディアコンテンツを製作する製作ステップを含むことを特徴するコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項13】 コンピュータに実行させるプログラムであって、

テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って生成されたマルチメディアコンテンツのフォーマットを、前記第1のフォーマットから第2のフォーマットに変換する変換ステップを含み、

前記変換ステップの処理により前記第2のフォーマットに変換された前記マルチメディアコンテンツは、

前記第2のフォーマットの前記マルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合されたマルチメディアコンテンツである

ことを特徴とするプログラム。

【請求項14】 前記変換ステップの処理により前記第2のフォーマットに変換された前記マルチメディアコンテンツと、前記ブラウザソフトウェアとを組み合わせ、1つのプログラムとする組み合わせステップをさらに含む

ことを特徴とする請求項13に記載のプログラム。

【請求項15】 前記第1のフォーマットで利用される前記技術は、XMLを利用する技術である

ことを特徴とする請求項13に記載のプログラム。

【請求項16】 コンピュータに実行させるプログラムであって、

テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って生成されたマルチメディアコンテンツのフォーマットを、前記第1のフォーマットから第2のフォーマットに変換する変換ステップを含み、

前記変換ステップの処理により前記第2のフォーマットに変換された前記マルチメディアコンテンツは、

前記第2のフォーマットの前記マルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、

前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合されたマルチメディアコンテンツである

ことを特徴するコンピュータが読み取り可能なプログラムが記録されている記録媒体。

【請求項17】 マルチメディアコンテンツを再生する情報処理装置において、

所定の前記マルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェアと、

前記ブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群、および、前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された



前記マルチメディアコンテンツと

を含むプログラムを取得する取得手段と、

前記取得手段により取得された前記ブラウザソフトウェアを実行する実行手段と、

前記実行手段により実行された前記ブラウザソフトウェアが提供するサービスを利用して、前記取得手段により取得された前記マルチメディアコンテンツを再生する再生手段と

を備えることを特徴とする情報処理装置。

【請求項18】 前記ブラウザソフトウェアを記憶する記憶手段をさらに備え、前記記憶手段は、前記ブラウザソフトウェアを予め記憶しており、

前記実行手段は、前記取得手段により前記マルチメディアコンテンツが取得された場合、前記記憶手段に予め記憶されている前記ブラウザソフトウェアを実行する

ことを特徴とする請求項17に記載の情報処理装置。

【請求項19】 マルチメディアコンテンツを再生する情報処理装置の情報処理方法において、

所定の前記マルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサー

ビスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェアと、

前記ブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群、および、前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された前記マルチメディアコンテンツと

を含むプログラムを取得する取得ステップと、

前記取得ステップの処理により取得された前記ブラウザソフトウェアを実行する実行ステップと、

前記実行ステップの処理により実行された前記ブラウザソフトウェアが提供するサービスを利用して、前記取得手段により取得された前記マルチメディアコンテンツを再生する再生ステップと

を含むことを特徴とする情報処理方法。

【請求項 2 0】 マルチメディアコンテンツを製作する情報処理装置において、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された前記マルチメディアコンテンツを製作する

ことを特徴とする情報処理装置。

【請求項 21】 マルチメディアコンテンツを製作する情報処理装置の情報処理方法において、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、  
複数個集合された前記マルチメディアコンテンツを製作する

ことを特徴とする情報処理方法。

【請求項 2 2】 マルチメディアコンテンツを製作する情報処理装置において、

テキストによりデータ構造を表現する技術を利用する第 1 のフォーマットに従って、前記マルチメディアコンテンツを製作する製作手段と、

前記製作手段により製作された前記マルチメディアコンテンツのフォーマットを、前記第 1 のフォーマットから第 2 のフォーマットに変換する変換手段とを備え、

前記変換手段により前記第 2 のフォーマットに変換された前記マルチメディアコンテンツは、

前記第 2 のフォーマットの前記マルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の 3D 仮想空間に表示させ、かつ、前記 3D 仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任

意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、  
複数個集合されたマルチメディアコンテンツである

ことを特徴とする情報処理装置。

【請求項 2 3】 前記変換手段により前記第 2 のフォーマットに変換された  
前記マルチメディアコンテンツと、前記ブラウザソフトウェアとを組み合わせ  
て、1 つのプログラムとする組み合わせ手段

をさらに備えることを特徴とする請求項 2 2 に記載の情報処理装置。

【請求項 2 4】 前記第 1 のフォーマットで利用される前記技術は、XML を  
利用する技術である

ことを特徴とする請求項 2 2 に記載の情報処理装置。

【請求項 2 5】 マルチメディアコンテンツを製作する情報処理装置の情報  
処理方法において、

テキストによりデータ構造を表現する技術を利用する第 1 のフォーマットに従  
って、前記マルチメディアコンテンツを製作する製作ステップと、

前記製作ステップの処理により製作された前記マルチメディアコンテンツのフ  
ォーマットを、前記第 1 のフォーマットから第 2 のフォーマットに変換する変換  
ステップと

を含み、

前記変換ステップの処理により前記第 2 のフォーマットに変換された前記マル  
チメディアコンテンツは、

前記第 2 のフォーマットの前記マルチメディアコンテンツを読み込み、前記  
マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポー  
ネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理  
するサービスと、

前記各コンポーネントのそれぞれを、所定の 3D 仮想空間に表示させ、かつ、  
前記 3D 仮想空間に既に表示されている前記コンポーネントを消去させるサービ  
スと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合されたマルチメディアコンテンツである

ことを特徴とする情報処理方法。

【請求項 26】 マルチメディアコンテンツを製作し、かつ、製作した前記マルチメディアコンテンツを配信する第1の情報処理装置と、

前記第1の情報処理装置により配信された前記マルチメディアコンテンツを再生する第2の情報処理装置とが、

所定のネットワークを介して相互に接続された情報処理システムにおいて、

前記第1の情報処理装置は、

所定のマルチメディアコンテンツを読み込み、前記マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、

前記コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、

前記各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、前記3D仮想空間に既に表示されている前記コンポーネントを消去させるサービスと、

前記スクリプトのインタプリタと、前記各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、

前記各コンポーネントのうちの所定のものと、他の前記コンポーネントとの間のフォーカス遷移を管理するサービスと、

所定の外部キーからの入力を、対応する前記コンポーネントに配布するサービスと

を前記各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、前記各コンポーネントのそれぞれに前記各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数の前記コンポーネントを、カスタマイズして任意に組み合わせた前記コンポーネント群と、

前記コンポーネントを操作するための前記スクリプトとを含む前記シーンが、複数個集合された前記マルチメディアコンテンツを製作し、

製作した前記マルチメディアコンテンツと、前記ブラウザソフトウェアを1つのプログラムとして、前記第2の情報処理装置に前記ネットワークを介して配信し、

前記第2の情報処理装置は、配信された前記プログラムのうちの前記ブラウザソフトウェア上に、前記プログラムのうちの前記マルチメディアコンテンツを再生させる

ことを特徴とする情報処理システム。

#### 【発明の詳細な説明】

##### 【0001】

#### 【発明の属する技術分野】

本発明は、プログラム、記録媒体、情報処理装置および方法、並びに、情報処理システムに関し、特に、高機能なマルチメディアコンテンツを容易に取り扱うことができるようにしたプログラム、記録媒体、情報送信装置および方法、並びに、情報処理システムに関する。

##### 【0002】

#### 【従来の技術】

近年、情報ネットワーク、中でも、インターネットの普及は、目覚ましいものがある。特に、W.W.W (World Wide Web) は、ビジネスのみならず家庭における一般ユーザ向けサービスとして欠かせないものになっている。W.W.Wは、インタ

ーネット上に存在する様々なコンテンツ（ホームページ）に受信端末がアクセスし、必要な情報を得ることを可能にするサービスである。

【0003】

近年、一般ユーザも、W.W.Wを単に情報収集のためだけではなく、エンターテインメント性の高いコンテンツを楽しんだり、また自らがコンテンツを発信したりするためのツールとして利用することが多くなっている。W.W.Wをこのように利用するユーザの数は、今後ますます増加するものと思われる。

【0004】

W.W.Wは、1989年にスイスのCERN（ヨーロッパ素粒子物理学研究所）で開発されたソフトウェア・アーキテクチャである。W.W.Wは、インターネットで結ばれたHTTP（HyperText Transfer Protocol）サーバ、およびHTML（HyperText Markup Language）ブラウザと称されるソフトウェア群から構成される通信システムであり、これを支えるソフトウェアがHTTPとHTMLである。HTTPは、HTTPサーバとHTMLブラウザとの間の通信プロトコルであり、HTMLは、HTTPサーバとHTMLブラウザとの間で交換される（ホームページの）データのフォーマットである。

【0005】

図1は、このようなW.W.Wのコンテンツ（ホームページ）再生の仕組みを表している。

【0006】

図1に示されるように、W.W.Wシステム1においては、HTMLブラウザ22、および、所定のOS（Operating System）やハードウェア等からなるプラットフォーム21から構成される受信端末11と、HTTPサーバ13とが、インターネット12を介して相互に接続されている。

【0007】

受信端末11がHTTPサーバ13上のコンテンツを再生するための処理について説明する。

【0008】

受信端末11のHTMLブラウザ22は、矢印31に示されるように、ホームページのURL（Uniform Resource Locator）がユーザにより入力されると、矢印32



に示されるように、入力されたURLで指定されたHTTPサーバ13にインターネット12を介してアクセスし、URLで指定されたホームページのデータを要求する。

#### 【0009】

HTTPサーバ13は、その要求に対して、HTMLフォーマットで記述されたホームページのデータファイル（以下、HTML文書と称する）33-1をインターネット12を介して受信端末11に送信する。

#### 【0010】

受信端末11のHTMLブラウザ22は、受信したHTML文書33-1を、例えば、受信端末11のディスプレイ（図示せず）上に表示34のような形で表示させる。また、HTMLブラウザ22は、必要に応じてビットマップファイル（画像ファイル）等の関連データ33-2を、HTTPサーバ13から取得し、表示34内の所定の領域に表示させる。

#### 【0011】

このように、HTMLブラウザ22は、一般ユーザがW.W.Wを利用する場合、欠かせないソフトウェアである。しかしながら、HTMLは、本来、研究者が情報ネットワークを介して技術文書などの文書データを共有するために開発されたものであり、マークアップという手法を用いて文書を構造化する点が大きな特徴である。

#### 【0012】

図2は、そのようなHTML文書とその表示例を表している。

#### 【0013】

HTML文章33-1において、“<”と“>”で挟まれた部分は、タグと称され、そのタグが利用されて、マークアップが行われる。

#### 【0014】

HTMLタグ（<HTML>及び</HTML>）で挟まれた部分は、その中身がHTMLのデータであることを表している。BODYタグ（<BODY>及び</BODY>）で挟まれた部分は、文書の本文であることを表している。H1タグ（<H1>及び</H1>）で挟まれた“<H1>鎌倉のちょっとおいしいお店</H1>”の部分41は、文字列「鎌倉のちょっとおいしいお店」が文書の見出しであることを表している。Pタグ（<P>及び</P>）で

挟まれた部分 4 2 と 4 3 は、文章の段落であることを表している。なお、このようなタグの種類の標準化は W3C (World Wide Web Consortium) で行われている。

## 【 0 0 1 5 】

このような HTML 文章 3 3 - 1 の文書構造に合わせてコンテンツの提示が行われたものが、表示 3 4 である。

## 【 0 0 1 6 】

表示 3 4 の上方の領域 5 1 には、H1 タグで挟まれた部分 4 1 に対応する文字列が、その下方の領域 5 2 には、P タグで挟まれた部分 4 2 に対応する文章が、さらにその下方には、P タグで挟まれた部分 4 3 に対応する文章が、それぞれ表示される。

## 【 0 0 1 7 】

このように HTML 文章 3 3 - 1 はテキストで表現されるので、HTML は、OS などのプラットフォームに依存しない。また、HTML は、テキストフォーマットであるので、ユーザ等は、多少の学習をすることにより、HTML を利用して、比較的容易にコンテンツを製作することができる。

## 【 0 0 1 8 】

さらに、HTML は、<IMG> タグを用いることで文書に画像を挿入することが出来る。例えば、表示 3 4 の領域 5 2 内の領域 5 4 には、HTML 文章 3 3 - 1 の P タグで挟まれた部分 4 2 内の <IMG> タグではさまれた箇所 4 4 に対応する画像ファイル (food.jpg) の画像 3 3 - 2 が表示される。

## 【 0 0 1 9 】

このように、HTML にはテキスト以外のメディアを表現するためのタグが用意されているので、HTML は、テキスト、画像 (静止画および動画)、並びに、音声といった複数の種類のメディアを組み合わせたコンテンツを取り扱うことが可能である。このようなコンテンツを、以下、マルチメディアコンテンツと称する。従って、いわゆるホームページとして知られる W.W.W のデータの多くも、マルチメディアコンテンツである。

## 【 0 0 2 0 】

このように、HTML は、プラットフォームに依存しないこと、マルチメディア情

報を扱えること、および、比較的容易にコンテンツを製作できることを特徴としている。

【0021】

また、インターネット以外の、デジタル放送や携帯電話網等のデジタル家電製品向け情報ネットワークにおいても、マルチメディアコンテンツを配信するサービスが広まりつつあり、HTMLの派生技術が採用されている。例えば、BS (Broadcasting Satellite) デジタル放送のデータ放送サービスで採用されたBML (Broadcast Markup Language) や、NTTドコモ社の携帯電話のサービスであるiモードで利用されているCompact HTML等が知られている。ただし、これらのサービスは、現状、特定のデジタル家電製品、または、特定の情報ネットワークのみでの利用が前提となっている場合が多く、コンテンツの相互流通も出来ないか、または制限されたものである。

【0022】

【発明が解決しようとする課題】

将来、コンピュータを含む全てのデジタル家電製品が1つの情報ネットワークで結ばれるようになることが想定されており、この場合、どの端末からでも同一のコンテンツによるサービスが受けられることが求められ、かつ、コンテンツの内容についても、エンターテイメント性が高く、より高機能、高性能、かつ低コストで開発が可能なものが求められるようになると想定される。

【0023】

しかしながら、上述したように、HTMLは、本来、研究者が文書データを共有するために開発されたものであり、上述したような、将来必須となると想定されるニーズを満たすためのシステムとして開発されたものではない。従って、HTMLは、このようなニーズを満たすソリューションとしては、以下のような課題を有している。

【0024】

即ち、上述したように、HTMLは構造化された文書、すなわちテキストというメディアを表現するためのフォーマットである。従って、HTMLにおいては、テキスト以外の他のメディア(画像等)は、そのテキスト(文章)中に挿入されるとい

う形態を取っている。

【0025】

しかしながら、このように、他のメディアが文書中に挿入されると、提示の一意性が保証されないという第1の課題がある。

【0026】

即ち、テキストに挿入されるメディアの位置は、そのテキストのレイアウトにより決定されるが、テキストはウィンドウの端まで行くと折り返される（例えば、図2に示されるように、表示34の領域52内においては、「パスタ屋さん」で折り返され、それに続く「をみつけました」は次の行に表示される）ので、HTMLブラウザのウィンドウのサイズが決定されるまで（ユーザがコンテンツを表示するまで）、そのメディアの表示位置は決定されないことになる。

【0027】

図3に示される表示34-1と図4に示される表示34-2は、このような、同一のHTML文章でも、メディアの表示位置がウィンドウのサイズの違いにより異なってしまうプレゼンテーションの例、即ち、提示の一意性が保証されないプレゼンテーションの例を表している。

【0028】

換言すると、HTMLによる表示のレイアウト方法は、文書の一部として画像等のメディアを表示する方法としては好適であるが、マルチメディアコンテンツとして、様々なメディアを組み合わせる方法としては好適であるとはいえない。

【0029】

なぜならば、マルチメディアコンテンツではデザインが重視されることが多く、コンテンツの製作者は、提示する個々のメディアの位置を絶対値で指定でき、HTMLブラウザのウィンドウサイズに関わらず、コンテンツを同一のデザインで提示することを望むことが多いからである。即ち、コンテンツの製作者は、自分自身が製作したマルチメディアコンテンツが、HTMLブラウザのウィンドウサイズによって、表示34-1（図3）または表示34-2（図4）のように可変されて表示されることは望んでおらず、例えば、図5の表示34-3に示されるように

、見出しのテキスト、2つの本文のテキスト、および画像のそれぞれが、表示34-3の固定された座標系上の絶対位置（図5の例では、左上端からの絶対値）で指定されて表示されることを望んでいる。

#### 【0030】

また、HTMLフォーマットはテキストの構造のみが規定されているので、HTMLデータが実際に表示される場合、テキストのフォント種類、字間、および行間などのレイアウトの詳細、並びに、ハイパーリンクの表現方法（青色のフォントと下線で表現される場合が多いが、規定はされていない）等は、HTMLブラウザに基本的に依存する。異なるベンダーにより開発されたHTMLブラウザのそれぞれにより、同一のHTML文書が提示されると、微妙に異なって表示される場合がある。これも提示の一意性が保証されていないことの現われである。

#### 【0031】

こうした状況に対して、HTML文書のスタイルを指定するための機構として、CSS (Cascading Style Sheets) が提案されている。CSSを利用することによりテキストのレイアウトを一意に指定することが可能である。しかしながら、CSSはあくまで文書のレイアウトを行うためのものであり、各社HTMLブラウザのサポートもまちまちで、また、後述するような3D (3 Dimensional) のプレゼンテーションにも対応していない。また、CSSは (HTMLとは異なる) 独自の文法を有しており、コンテンツ製作者はそれに習熟する必要がある。

#### 【0032】

このように、HTMLは、マルチメディア表現の必須条件である、提示の一意性が保証されないという第1の課題を有している。

#### 【0033】

さらに、マルチメディア表現という観点から見て、HTMLは、メディア間の同期機構が備わっていないという第2の課題を有している。

#### 【0034】

メディア間の同期機構は、音声、動画、およびアニメーションなどのように、時間軸を持つメディアが組み合わされて提示される場合、それらの提示のタイミングを制御する仕組みである。メディアの同期機構を利用することで、例えば、

音楽に合わせたアニメーションの再生や、複数の動画の同時または連続した再生などといった表現が可能になる。このように、メディアの同期機構は、マルチメディアコンテンツの実現には欠かせない機能である。

【 0 0 3 5 】

また、HTMLは、そのコンテンツの、異種の機器間（例えば、パーソナルコンピュータとデジタル家電製品の間）における互換性がないという第3の課題を有している。

【 0 0 3 6 】

即ち、HTMLには、コンテンツを操作するためのインタフェースに関する規定はないが、パーソナルコンピュータ上のHTMLブラウザの多くは、パーソナルコンピュータのインタフェース、例えば、ウィンドウズ（登録商標）OSが提供するウィンドウ上に実現される。

【 0 0 3 7 】

このウィンドウはサイズが可変とされるので、HTMLブラウザのサイズも可変とされる。即ち、ブラウザの横方向のサイズは不定であり、表示しきれないコンテンツは下へはみ出して、表示されない。これを表示するために縦方向のスクロールが利用される。また、ハイパーリンクによる他のHTML文書へのジャンプ等のコンテンツ操作には、マウスが利用される。コンテンツもこのようなパーソナルコンピュータ依存の操作性が、前提とされて製作される。

【 0 0 3 8 】

これに対して、デジタル家電製品においても、HTML文章の表示が可能であるが、HTMLブラウザの表示サイズは一般的に固定されており、またスクロール機能が無い場合もある。さらに、マウス等のムービングカーソルの操作は慣れが必要であり、家電製品には適さないと考えられているので、マウスに相当する機能も無い場合が多い。従って、デジタル家電製品にHTMLが適用される場合、デジタル家電製品のインタフェースで利用可能とされるように、HTMLフォーマットに対して何らかの変更又は拡張がなされるのが一般的である。

【 0 0 3 9 】

このようなパーソナルコンピュータとデジタル家電製品の違いにより、インタ

ーネット上で配信されているHTMLコンテンツが、デジタル家電製品のネットワーク上で配信された場合、互換性が保てない可能性がある。この差異を吸収する変換ソフトウェアが一部で利用されているが、インターネット上のHTMLコンテンツは、上述したようにパーソナルコンピュータのインタフェースに基づいて製作されているので、そのような変換ソフトウェアにより変換されたHTMLコンテンツがデジタル家電製品で仮に再生されても、それは、意味をなさないプレゼンテーションになる可能性がある。

## 【0040】

このように、HTMLは、そのコンテンツの互換性がないという第3の課題を有している。

## 【0041】

さらに、HTMLは、その機能拡張に限界があるという第4の課題を有している。

## 【0042】

即ち、HTMLは、上述したように、本来、構造化された文書データであり、アプリケーションプログラムではない。従って、HTMLブラウザは、HTTPサーバに予め用意されたデータを表示することしかできない。また、表示可能なデータの種類（メディア）は、HTMLのフォーマットで規定されたものに限られる。

## 【0043】

例えば、新規に開発されたメディアが利用される場合、上述したW3Cで標準化されたHTMLのフォーマットに対して拡張が行われる必要がある。また、そのメディアが利用されるためには、各社のHTMLブラウザが、その新規メディアを再生するためのソフトウェア・モジュールをサポートしており、拡張した部分、すなわちタグを正しく認識する必要がある。

## 【0044】

しかしながら、実際には、各社のHTMLブラウザで独自にタグの拡張がなされている場合が多く、これらのブラウザ間でコンテンツの互換性がない場合がある。またHTMLコンテンツも特定のHTMLのブラウザの、特定の拡張に依存している場合がある。

## 【0045】

HTMLコンテンツの機能を拡張する他の方法として、プラグイン機構がある。プラグインは、HTMLブラウザに対する機能拡張用のソフトウェア・モジュールである。任意の機能を実現するプラグインが開発され、HTMLブラウザにインストールすることで、HTMLコンテンツからその機能を実行させることができる。しかしながら、ユーザは、このようなプラグインを利用するために、そのダウンロードやインストール、アップグレードを事前に行わなければならない。またプラグインは、OSやブラウザ毎に開発される必要があり、そのプラグインをサポートしていないHTMLブラウザでは、そのプラグインを利用したHTMLコンテンツは表示できない。

## 【0046】

さらに、正確には、プラグインによる拡張機能は、HTMLコンテンツの一部の領域で他のアプリケーションが動作するという形態をとっており、コンテンツ自体にその機能が含まれている訳ではなく、HTML自体は構造化された文書であることに変わりがない。

## 【0047】

なお、HTML文書自体に任意のロジックを実行させることを目指した拡張としてJava(R)Scriptがある。Java(R)Scriptは、スクリプト言語であり、C言語やJava(R)言語と同様に、条件分岐や繰り返しといった制御構造をサポートしながら、デザイナーやコンテンツクリエイターといったコンテンツ製作者でも比較的容易に記述できるように考慮された文法を有している。またJava(R)Scriptのコードはインタプリタで実行するので、事前にコンパイルする必要がない、といった扱いやすさが特徴である。一方で、Java(R)Scriptは、扱いやすさを重視した代わりに、実行パフォーマンスやプログラムサイズ、またプログラムの構造化といった点ではC言語やJava(R)言語に劣る。

## 【0048】

このような理由から、Java(R)Scriptは、HTMLが提供するメディアやプラグイン、またブラウザそのものを操作するといった比較的単純な用途に適しているが、Java(R)Script自体で複雑な機能（例えば、特定のメディアのデコーダやGUI部品）を実現するのは現実的でない場合が多い。



## 【0049】

このように、HTMLは、その機能拡張に限界があるという第4の課題を有している。

## 【0050】

さらに、HTMLは、そのコンテンツの生産性が劣るという第5の課題を有している。

## 【0051】

即ち、上述したように、HTMLは、研究者が構造化された文書データを共有することを目的に開発されたものである。そのような用途に限って言えば、ソフトウェアの素養のある研究者にとっては、HTMLのフォーマットに従って、テキストエディタ等のツールでコンテンツを記述することは比較的容易な作業であるといえる。

## 【0052】

これに対して、デザイナーやクリエイターは、よりエンターテインメント性が高く複雑なコンテンツを製作する場合、テキストエディタ等を使って直接HTML文書を記述することは稀であり、一般的に、専用のオーサリング・ソフトウェアを利用する。それらのソフトウェアは、グラフィカルなコンテンツ製作環境を提供することにより、HTMLに関する知識が少なくとも、比較的複雑なレイアウトのコンテンツを製作することを可能とする。

## 【0053】

しかしながら、HTMLのフォーマット自体は、そのようなオーサリング・ソフトウェアとの親和性を考慮して設計されているわけではない。その弊害として、ソフトウェアが機械的にHTML文書を作成することによるデータサイズのオーバーヘッドや、オーサリング・ソフトウェア間の互換性、コンテンツの仕上げ段階の微妙な調整の難しさ、および、作成されたHTML文書の可読性の低さ等が挙げられ、HTMLの利点がむしろ欠点になっているような状況である。

## 【0054】

また、上述したようにHTMLコンテンツに拡張機能が組み込まれる場合、Java(R)Script等のスクリプト言語が利用されることが多いが、多少複雑な機能でもデ

デザイナーやコンテンツクリエイターが、自分自身でJava(R)Scriptのコードを記述する場合が多い。

## 【 0 0 5 5 】

しかしながら、コンテンツの生産性の観点からは、こうした拡張機能は、ソフトウェアに関してより高いスキルを持つプログラマにより、C言語やJava(R)言語等のプログラム言語が利用されて開発される方が好適である。また、コンテンツの生産性の観点から見て、デザイナーやクリエイターは、それらの拡張機能をどのようにカスタマイズし、組み合わせるか、また、どのようなサービスを実現するかに注力する方が好適であり、Java(R)Script等のスクリプト言語は、デザイナーやコンテンツクリエイターがこれらの拡張機能を組み合わせて制御するために利用される方が好適である。

## 【 0 0 5 6 】

しかしながら、現状では、HTMLコンテンツの製作において、そのような分業体制は確立されておらず、HTMLのフォーマット自体もこうした開発フローとの親和性を考慮して設計されているわけではない。

## 【 0 0 5 7 】

また、一度開発された拡張機能は、コンテンツの生産性の観点からは、再利用されるべきであるが、Java(R)Script等のスクリプト言語はソフトウェアのモジュール化に関する機能（カスタマイズのしやすさ、実装のカプセル化、ソフトウェア・モジュールの配布等）が十分にサポートされているとはいえず、再利用は困難である。

## 【 0 0 5 8 】

このように、HTMLは、そのコンテンツの生産性が劣るという第5の課題を有している。

## 【 0 0 5 9 】

また、HTMLは、ハードウェアの性能を十分に発揮させることが困難であるという第6の課題を有している。

## 【 0 0 6 0 】

即ち、パーソナルコンピュータを含め、デジタル家電製品のハードウェアの機

能、および性能は、今後ますます向上していくことと思われる。特に、近い将来、一部の家庭用ゲームで実現されているような、高性能な3Dグラフィックス機能が他の製品でも採用される可能性が大きい。

【 0 0 6 1 】

これに対して、HTMLは、本来、構造化された文書の共有を目的としたソフトウェアであり、3Dを前提としたコンテンツの表現には不適である。上述したように、プラグインやJava(R)Script等のプラグインによってHTMLのコンテンツに機能の拡張を行うことは可能であるが、それぞれ上述したような課題を抱えており、近い将来のデジタル家電製品が有するであろうハードウェア性能を引き出すためには、HTMLは、必ずしも適切なソリューションとなりうるものではない。

【 0 0 6 2 】

このように、HTMLは研究者の間で文書データを共有するというニーズを満たすために開発されたソフトウェア・アーキテクチャである。文書データの共有という観点からは、現在でも最も洗練されたソリューションの1つであると考えられる。しかしながら、現在の情報ネットワーク上でのコンテンツ配信におけるニーズは、高機能指向、マルチメディア指向であり、かつデザイナーやコンテンツクリエイターといった職種のコンテンツ製作者にとって製作が容易であることが求められている。

【 0 0 6 3 】

このようなニーズの変化に対応するため、これまでHTMLに様々な機能の拡張がなされてきたが、上述したように、HTML本来の設計思想に沿ったものでないため、コンテンツの互換性や生産性などの面で根本的な解決に至っておらず、上述した第1乃至第6の課題を抱えている。

【 0 0 6 4 】

本発明は、このような状況に鑑みてなされたものであり、高機能なマルチメディアコンテンツを容易に取り扱うことができるようにするものである。

【 0 0 6 5 】

【課題を解決するための手段】

本発明の第1のプログラムは、所定のマルチメディアコンテンツを読み込み、

マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェアと、ブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとを含むことを特徴とする。

## 【0066】

ブラウザソフトウェアは、さらに、複数のメディアを同期再生するサービスと、第1のシーンと、第2のシーンとの間を遷移する遷移サービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するようにすることができる。

## 【0067】

マルチメディアコンテンツは、ブラウザソフトウェア上で再生される場合、提示の一意性が保たれるように構成されるようにすることができる。

## 【0068】

本発明の第1の記録媒体のプログラムは、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、ス

クリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェアと、ブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとを含むことを特徴とする。

## 【 0 0 6 9 】

本発明の第 1 のプログラムおよび記録媒体においては、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェアと、そのブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、そのコンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとを含むプログラムが提供される。

## 【 0 0 7 0 】

本発明の第 2 のプログラムは、マルチメディアコンテンツとしてのプログラムであって、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の 3D 仮想空間に表示させ、かつ、3D 仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに

配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツを含むことを特徴とする。

## 【 0 0 7 1 】

本発明の第2のプログラムの記録媒体は、マルチメディアコンテンツとしてのプログラムであって、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツを含むことを特徴とする。

## 【 0 0 7 2 】

本発明の第2のプログラムおよび記録媒体においては、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェア上で動作する複数のコンポーネントが、カスタマイズして任意に組み合わせられたコンポーネント群と、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメデ

イアコンテンツが、プログラムとして提供される。

【0073】

本発明の第3のプログラムは、ブラウザソフトウェアとしてのプログラムであって、マルチメディアコンテンツとしてのプログラムであって、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供することを特徴とする。

【0074】

本発明の第3の記録媒体のプログラムは、ブラウザソフトウェアとしてのプログラムであって、マルチメディアコンテンツとしてのプログラムであって、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれにそれらのサービスを提供することを特徴とする。

## 【0075】

本発明の第3のプログラムおよび記録媒体においては、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェアが供給される。

## 【0076】

本発明の第4のプログラムは、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツを製作する製作ステップを含むことを特徴とする。

## 【0077】

製作されたマルチメディアコンテンツと、ブラウザソフトウェアとを組み合わせ、1つのプログラムにする組み合わせステップをさらに含むようにすることができる。

## 【0078】

製作ステップは、配布形式のコンポーネントを読み込んでGUI上に表示させるように制御する表示制御ステップと、GUI上で選択されたコンポーネントをマルチメディアコンテンツに追加する追加ステップと、マルチメディアコンテンツ上に含まれるコンポーネントのカスタマイズを行うGUIを提供する提供ステップと



、表示制御ステップ、追加ステップ、および、提供ステップの処理により編集されたマルチメディアコンテンツを、配布可能な形式に変換する変換ステップとを含むようにすることができる。

【0079】

本発明の第4の記録媒体のプログラムは、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツを製作する製作ステップを含むことを特徴とする。

【0080】

本発明の第4のプログラムおよび記録媒体においては、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツが製作されるプログラムが提供される。

【0081】

本発明の第5のプログラムは、テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って生成されたマルチメディアコンテンツのフォ

ーマットを、第1のフォーマットから第2のフォーマットに変換する変換ステップを含み、変換ステップの処理により第2のフォーマットに変換されたマルチメディアコンテンツは、第2のフォーマットのマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群と、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツであることを特徴とする。

#### 【0082】

変換ステップの処理により第2のフォーマットに変換されたマルチメディアコンテンツと、ブラウザソフトウェアとを組み合わせ、1つのプログラムとする組み合わせステップをさらに含むようにすることができる。

#### 【0083】

第1のフォーマットで利用される技術は、XMLを利用する技術であるようにすることができる。

#### 【0084】

本発明の第5の記録媒体のプログラムは、テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って生成されたマルチメディアコンテンツのフォーマットを、第1のフォーマットから第2のフォーマットに変換する変換ステップを含み、変換ステップの処理により第2のフォーマットに変換され

たマルチメディアコンテンツは、第2のフォーマットのマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群と、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツであることを特徴する。

## 【0085】

本発明の第5のプログラムおよび記録媒体においては、テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って生成されたマルチメディアコンテンツのフォーマットが、第1のフォーマットから第2のフォーマットに変換される。その第2のフォーマットに変換されたマルチメディアコンテンツは、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとされる。

## 【0086】

本発明のマルチメディアコンテンツを再生する装置としての情報処理装置は、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトの

それぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェアと、ブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとを含むプログラムを取得する取得手段と、取得手段により取得されたブラウザソフトウェアを実行する実行手段と、実行手段により実行されたブラウザソフトウェアが提供するサービスを利用して、取得手段により取得されたマルチメディアコンテンツを再生する再生手段とを備えることを特徴とする。

## 【 0 0 8 7 】

ブラウザソフトウェアを記憶する記憶手段をさらに設け、記憶手段は、ブラウザソフトウェアを予め記憶しており、実行手段は、取得手段によりマルチメディアコンテンツのみが取得された場合、記憶手段に予め記憶されているブラウザソフトウェアを実行するようにすることができる。

## 【 0 0 8 8 】

本発明のマルチメディアコンテンツを再生する装置としての情報処理装置の情報処理方法は、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、

各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェアと、ブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとを含むプログラムを取得する取得ステップと、取得ステップの処理により取得されたブラウザソフトウェアを実行する実行ステップと、実行ステップの処理により実行されたブラウザソフトウェアが提供するサービスを利用して、取得手段により取得されたマルチメディアコンテンツを再生する再生ステップとを含むことを特徴とする。

## 【 0 0 8 9 】

本発明のマルチメディアコンテンツを再生する装置としての情報処理装置および方法においては、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェアと、所定の構成を有するマルチメディアコンテンツとからなるプログラムが取得され、取得されたブラウザソフトウェアが実行され、実行されたブラウザソフトウェアが提供するサービスが利用されて、取得されたマルチメディアコンテンツが再生される。

## 【 0 0 9 0 】

本発明のマルチメディアコンテンツを製作する第1の装置としての情報処理装置は、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポ

ーメントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツを製作することを特徴とする。

## 【 0 0 9 1 】

本発明のマルチメディアコンテンツを製作する第1の装置としての情報処理装置の情報処理方法は、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツを製作することを特徴とする。

## 【 0 0 9 2 】

本発明のマルチメディアコンテンツを製作する第1の装置としての情報処理装置および方法においては、所定のサービスをコンポーネントが受けるために満た

すべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェア上で動作し、かつ、所定の構成を有するマルチメディアコンテンツが製作される。

【 0 0 9 3 】

本発明のマルチメディアコンテンツを製作する第2の装置としての情報処理装置は、テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って、マルチメディアコンテンツを製作する製作手段と、製作手段により製作されたマルチメディアコンテンツのフォーマットを、第1のフォーマットから第2のフォーマットに変換する変換手段とを備え、変換手段により第2のフォーマットに変換されたマルチメディアコンテンツは、第2のフォーマットのマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群と、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツであることを特徴とする。

【 0 0 9 4 】

変換手段により第2のフォーマットに変換されたマルチメディアコンテンツと、ブラウザソフトウェアとを組み合わせ、1つのプログラムとする組み合わせ手段をさらに設けるようにすることができる。

【 0 0 9 5 】

第1のフォーマットで利用される技術は、XMLを利用する技術であるようにすることができる。

【0096】

本発明のマルチメディアコンテンツを製作する第2の装置としての情報処理装置の情報処理法は、テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って、マルチメディアコンテンツを製作する製作ステップと、製作ステップの処理により製作されたマルチメディアコンテンツのフォーマットを、第1のフォーマットから第2のフォーマットに変換する変換ステップとを含み、変換ステップの処理により第2のフォーマットに変換されたマルチメディアコンテンツは、第2のフォーマットのマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群と、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツであるようにすることができる。

【0097】

本発明のマルチメディアコンテンツを製作する第2の装置としての情報処理装置および情報処理法においては、テキストによりデータ構造を表現する技術を利用する第1のフォーマットに従って、マルチメディアコンテンツが製作され、製作されたマルチメディアコンテンツのフォーマットが、第1のフォーマットから



第2のフォーマットに変換される。その第2のフォーマットに変換されたマルチメディアコンテンツは、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとされる。

## 【0098】

本発明の情報処理システムは、マルチメディアコンテンツを製作し、かつ、製作したマルチメディアコンテンツを配信する第1の情報処理装置と、第1の情報処理装置により配信されたマルチメディアコンテンツを再生する第2の情報処理装置とが、所定のネットワークを介して相互に接続された情報処理システムであって、第1の情報処理装置は、所定のマルチメディアコンテンツを読み込み、マルチメディアコンテンツを構成する各シーンのそれぞれに含まれる、コンポーネント群、および、スクリプトのそれぞれをロードするサービスと、コンポーネント群を構成する各コンポーネントのライフサイクルを管理するサービスと、各コンポーネントのそれぞれを、所定の3D仮想空間に表示させ、かつ、3D仮想空間に既に表示されているコンポーネントを消去させるサービスと、スクリプトのインタプリタと、各コンポーネントのそれぞれとの間の通信サービスを提供するサービスと、各コンポーネントのうちの所定のものと、他のコンポーネントとの間のフォーカス遷移を管理するサービスと、所定の外部キーからの入力を、対応するコンポーネントに配布するサービスとを各コンポーネントのそれぞれが受けるために満たすべき仕様に従って、各コンポーネントのそれぞれに各サービスのそれぞれを提供するブラウザソフトウェアと、ブラウザソフトウェア上で動作する複数のコンポーネントを、カスタマイズして任意に組み合わせたコンポーネント群、および、コンポーネントを操作するためのスクリプトとを含むシーンが、複数個集合されたマルチメディアコンテンツとを製作し、製作したマルチメディアコンテンツと、ブラウザソフトウェアを1つのプログラムとして、第2の情報処理装置にネットワークを介して配信し、第2の情報処理装置は、配信されたプログラムのうちのブラウザソフトウェア上に、そのプログラムのうちのマルチメデ

ィアコンテンツを再生させることを特徴とする。

【 0 0 9 9 】

本発明の情報処理システムにおいては、第 1 の情報処理装置により、所定のサービスをコンポーネントが受けるために満たすべき仕様に従って、コンポーネントにそれらのサービスを提供するブラウザソフトウェア上で動作し、かつ、所定の構成を有するマルチメディアコンテンツが製作され、製作されたマルチメディアコンテンツと、ブラウザとからなる 1 つのプログラムが配信され、第 2 の情報処理装置により、配信されたブラウザソフトウェア上に、配信されたマルチメディアコンテンツが表示される。

【 0 1 0 0 】

本発明のプログラムを実行するコンピュータは、パーソナルコンピュータ等の単独のコンピュータでもよいし、デジタル家電製品等の他の装置に組み込まれたコンピュータでもよい。同様に、本発明の情報処理装置は、パーソナルコンピュータ等でもよいし、デジタル家電製品等の他の装置でもよい。

【 0 1 0 1 】

【発明の実施の形態】

はじめに、本実施の形態が適用する基礎技術の 1 例である Java(R) について説明する。

【 0 1 0 2 】

Java(R) は、汎用的なプログラム開発または実行のための基盤ソフトウェアであり、情報ネットワーク上でのソフトウェア配信に適したソフトウェアとしてサン・マイクロシステムズ社によって開発されたものである。

【 0 1 0 3 】

Java(R) テクノロジーの中核は、次の (1) 乃至 (4) に示されるものである。

【 0 1 0 4 】

- (1) Java(R) 言語と称されるプログラム言語仕様
- (2) Java(R) 言語で記述されたプログラムを、中間コードと称されるフォーマットに変換するためのコンパイラ

(3) 中間コードを、コンピュータ上で実行するためのソフトウェアである Java(R)バーチャルマシン

(4) Java(R)言語用に標準化されたクラスライブラリ

【0105】

特に、バーチャルマシンと標準化クラスライブラリを併せて、Java(R)実行環境と称している。

【0106】

中間コードとライブラリの仕様は、OS等のプラットフォームに非依存となるように設計されているので、結果として、標準化ライブラリのみが外部モジュールとして利用されて、開発されたJava(R)プログラム（実際には、それが中間コードにコンパイルされたもの）は、端末側にJava(R)実行環境がサポートされていれば、その下位のプラットフォームのアーキテクチャに関わらず実行可能である。また標準化ライブラリ以外の外部モジュールでも、Java(R)言語で開発されていて、各プラットフォームに存在するものであれば利用可能である。

【0107】

さらに、Java(R)の標準化ライブラリには、URLを指定することで、HTTP等のプロトコルを介してネットワークから特定のファイルをダウンロードする機能等も用意されている。これにより、情報ネットワークからJava(R)アプリケーションを取得、実行することが可能になる。

【0108】

このようなJava(R)の特徴は、ネットワークを介したコンテンツ配信に大きな可能性をもたらす。なぜならば、パーソナルコンピュータやデジタル家電製品などの端末にJava(R)の実行環境を搭載することにより、従来の、HTMLのようなフォーマットに基づいたデータの配信ではなく、それ自体がロジックを備えたアプリケーションの配信をクロスプラットフォームで行うことが可能になるからである。

【0109】

図6と図7は、HTML文章と、Java(R)アプリケーションの配信の方式の違いを表している。即ち、図6は、HTMLコンテンツの配信の方式を表しており、これに

対して、図7は、Java(R)アプリケーションの配信の方式を表している。

【0110】

図6に示されるように、HTMLコンテンツの配信の場合、ネットワークを介して配信されるものは、配信コンテンツ61であり、具体的には、HTML文書63、即ち、データである。また、端末ソフトウェア62は、HTML文章63の解釈、および、再生を行うHTMLブラウザ64、並びに、OS等のプラットフォーム63から構成される。従って、上述したように、そのコンテンツにより実現できる機能は、HTMLブラウザの機能に限定される。

【0111】

一方、図7に示されるように、Java(R)アプリケーションの配信の場合、ネットワークを介して配信されるものは、配信アプリケーション71、即ち、Java(R)アプリケーション73である。端末ソフトウェア72は、Java(R)アプリケーション73を実行するJava(R)実行環境74、および、プラットフォーム75である。従って、Java(R)言語で実装できる機能であればどのようなものでも配信コンテンツとして実現可能となる。

【0112】

実際に、欧州の標準化団体DVB (Digital Video Broadcasting) によって規格化が行われたデジタル家電製品用ソフトウェアプラットフォームの規格であるMHP (Multimedia Home Platform) では、このような動向に先がけて、Java(R)言語で作成されたアプリケーションの実行環境のサポートが盛り込まれている。

【0113】

ただし、Java(R)は汎用的な用途で用いられるソフトウェア環境である。Java(R)のコンテンツを開発するには、Java(R)言語によるプログラミングのスキルが必要であり、デザイナーやコンテンツクリエイターが直接Java(R)言語を利用してコンテンツを開発するのは現実的でないし、そうあるべきでもない。さらに、コンテンツの開発コストもHTMLの場合と比べて大きくなる場合もある。

【0114】

従って、上述した第1乃至第6の課題についても、Java(R)を採用したというだけで解決されるわけではない。即ち、Java(R)のもつ特長を利用して、これら

の課題をいかにして解決するかが重要なことである。

【0115】

そこで、本実施の形態においては、以下の特徴を有するマルチメディアコンテンツの製作、配信、および、再生を可能にするソフトウェアを実現することによって、上述した課題を解決している。

【0116】

第1の特徴として、マルチメディアに適したプレゼンテーションおよび、操作性を実現可能にすることが挙げられる。具体的には、次に示す(11)乃至(15)のような特徴がある。

【0117】

(11) 提示の一意性の保証

本実施の形態のソフトウェアにおいては、個々のモノメディアを画面上に配置していくことによってマルチメディアコンテンツを製作する。実際には、各種のモノメディアのそれぞれは、後述するコンポーネントの一種として実現される。本実施の形態のソフトウェアによるコンテンツのレイアウトは、そのコンテンツを構成する各コンポーネントの位置を画面上の絶対座標で指定することによって行われる。これにより、製作したコンテンツは再生時に、プラットフォームの種類、またはブラウザのウィンドウサイズに関わらず、製作時と同じレイアウト及びデザインで提示される。

【0118】

(12) 上下左右及び決定キーによるナビゲーション

本実施の形態のソフトウェアにおいては、ユーザとコンテンツとのインタフェースとして、(上下左右の)矢印キーと決定キーが利用される。本実施の形態のソフトウェアを実行するプラットフォームが、これらのキーをハードウェア的に提供することを想定している。図8は、これらのハードウェアとしてのキー群の例を表している。即ち、本実施の形態のユーザとコンテンツとのインタフェースは、決定キーとしてのキー91-1、上矢印キーとしてのキー91-2、右矢印キーとしてのキー91-3、下矢印キーとしてのキー91-4、および、左矢印キーとしてのキー91-5から構成される。このようなキー群91を有する入力

デバイスの代表的なものとして、リモートコントローラ（以下、単にリモコンと称する）が挙げられる。また、これらのキー群91を有する入出力デバイスは、リモコンに限定されず、パーソナルコンピュータはもちろん、モバイル端末や各種デジタル家電製品においても、同一の操作でコンテンツとのインタラクションを行うことが可能とされている。ユーザは、このボタン群91を利用して、画面上のコンポーネントの1つを選択することにより、コンテンツとのインタラクションを行う。

#### 【0119】

##### （13）シーン単位のコンテンツ構成

本実施の形態のソフトウェアにおけるマルチメディアコンテンツは複数のシーン（画面）から構成され、シーン単位で配信される。

#### 【0120】

##### （14）メディア同期機構

本実施の形態は、動画や音声、アニメーション等、時間軸を持つ複数のメディアを同期させて提示する機能を提供する。

#### 【0121】

##### （15）3Dベースのプレゼンテーション

本実施の形態におけるマルチメディアコンテンツは、3Dベースのプレゼンテーションを基本とする。これにより、直感的で分かり易いユーザーインタフェースの構築が可能となっている。

#### 【0122】

本実施の形態のソフトウェアの第2の特徴として、低コストで、高機能なコンテンツ製作を実現可能にするコンポーネントフレームワークが挙げられる。具体的には、次に示す（21）乃至（24）のような特徴がある。

#### 【0123】

##### （21） 任意のコンポーネントを組み合わせてコンテンツを構成する

本実施の形態のマルチメディアコンテンツにおいては、各シーンは、任意のコンポーネントの組み合わせから構成される。コンポーネントの例として、動画若しくは静止画などのモノメディア部品、または、ボタンなどのGUI（Graphical U

ser Interface) 部品が挙げられる。実際のコンテンツ製作は、後述するようなオーサリングシステムを利用して行われる。コンテンツ製作者は、オーサリングシステム上でシーンにコンポーネントを追加し、コンポーネントのカスタマイズを行ったり、スクリプト言語を利用してその振る舞いを記述したりすることが出来る。

#### 【0124】

(22) ブラウザソフトウェア（以下、単にブラウザと称する）によるコンテンツの再生

本実施の形態のマルチメディアコンテンツは、ブラウザによって再生される。ブラウザは、指定されたシーンをロードし、シーンを構成するコンポーネント群の読み込み及び3D表示を行う。さらにブラウザは、コンポーネントのライフサイクルの管理、キー入力によるナビゲーションのサポート、スクリプトとコンポーネントのバインディング等のサービスもコンポーネントに提供する。

#### 【0125】

(23) コンポーネントのインタフェース仕様

コンポーネントは、ブラウザのサービスを受けるためには、定められたインタフェース仕様を満たした実装をする必要があるが、それ以外はソフトウェア的にコンポーネントとブラウザは完全に分離されている。従って、その仕様を満たしていれば、任意の機能を持ったコンポーネントを自由に新規開発し、コンテンツに組み込むことが可能である。

#### 【0126】

(24) スクリプト言語によるコンポーネント制御

本実施の形態のソフトウェアにおいては、スクリプト言語を採用することにより、デザイナーやコンテンツクリエイターといった、専門的なプログラム言語のスキルがないと考えられる一般のコンテンツ製作者でも、コンポーネントを操作したり、コンポーネントからのイベントを受け取って、その種類に応じた処理を行ったりという、コンポーネントの振る舞いの制御を記述することを可能としている。

#### 【0127】

本発明のソフトウェアの第 3 の特徴として、クロスプラットフォームのコンテンツ再生環境が挙げられる。具体的には、次に示す (3 1) と (3 2) のような特徴がある。

【 0 1 2 8 】

( 3 1 ) ブラウザの配信

ブラウザはコンテンツとともに配信することが可能である。これにより、ブラウザのバージョンアップや他のアーキテクチャとの共存が容易になる。

【 0 1 2 9 】

( 3 2 ) クロスプラットフォーム

配信されるコンテンツ及びブラウザは、Java(R)等のクロスプラットフォームで動作する言語で作成されたアプリケーションであり、その動作環境が提供されるプラットフォームであれば実行が可能である。

【 0 1 3 0 】

以下、Java(R)を利用した場合の本実施の形態を説明する。

【 0 1 3 1 】

図 9 は、本実施の形態が適用された情報配信システムの構成例を表している。

【 0 1 3 2 】

図 9 に示されるように、情報配信システム 1 0 1 においては、サービス事業者が管理するサービス事業者装置群 1 1 1 (具体的には、後述するサーバ 1 2 4) と、ユーザが管理する受信端末 1 1 3 が、所定のネットワーク 1 1 2 を介して相互に接続されている。

【 0 1 3 3 】

サービス事業者装置群 1 1 1 は、モノメディア製作装置 1 2 1、コンポーネント開発装置 1 2 2、マルチメディアコンテンツ製作装置 1 2 3、および、サーバ 1 2 4 から構成されている。なお、この例においては、モノメディア装置 1 2 1 乃至サーバ 1 2 4 のそれぞれは別の装置とされているが、モノメディア装置 1 2 1 乃至サーバ 1 2 4 は、1 つの装置に集約されてもよいし、所定の個数の装置 (例えば、モノメディア製作装置 1 2 1 乃至マルチメディアコンテンツ製作装置 1 2 3 の機能を有する第 1 の装置と、サーバ 1 2 4 の機能を有する第 2 の装置との



2つの装置)に分割されてもよい。また、その分割の区切りも任意である。

【0134】

なお、サービス事業者装置群111を管理するものは、特に限定されないが、マルチメディアコンテンツを製作し、ネットワーク112を介してユーザ(受信端末113)に配信する主体である。現在のインターネットにおけるW.W.Wのように、営利若しくは非営利を目的とする個人、または、団体など様々な形態の主体が考えられる。

【0135】

モノメディア製作装置121は、動画、静止画、音声、テキスト、または、3D形状など、マルチメディアコンテンツに組み込むための各種モノメディアを製作し、マルチメディアコンテンツ製作装置123に供給する。モノメディア製作装置121は、一般的には、デザイナーにより操作される。

【0136】

コンポーネント開発装置122は、本実施の形態におけるコンポーネントを開発し、マルチメディアコンテンツ製作装置123に供給する。コンポーネントはJava(R)言語により開発されるので、コンポーネント開発装置122は、一般的には、プログラマにより操作される。また、既に存在するコンポーネントのみでマルチメディアコンテンツの製作が可能な場合は、コンポーネント開発装置122は省略されてもよい。

【0137】

マルチメディアコンテンツ製作装置123は、コンポーネント開発装置122より供給された各種コンポーネントを組み合わせるマルチメディアコンテンツを製作し、さらに、そのマルチメディアコンテンツに、モノメディア製作装置121より供給された各種のモノメディアを組み込み、サーバ124に供給する。

【0138】

さらにまた、マルチメディアコンテンツ製作装置123は、必要に応じて、製作したマルチメディアコンテンツと、それを再生するためのブラウザとからなるソフトウェアを製作し、サーバ124に供給する。

【0139】

具体的には、マルチメディアコンテンツ製作装置123には、後述するようなオーサリングシステムがインストールされており、コンテンツ製作者は、このようなシステムを利用して、コンポーネントをコンテンツに組み込み、カスタマイズして、スクリプト言語によりその振る舞いを記述する。この場合、コンテンツ製作者は、Java(R)言語の知識を有する必要はない。従って、マルチメディアコンテンツ製作装置123は、一般的に、Java(R)言語の知識を特に有しないデザイナーやコンテンツクリエイターにより直接操作される。

## 【0140】

サーバ124は、ネットワーク112と接続されており、マルチメディアコンテンツ製作装置123より供給されたマルチメディアコンテンツを、ネットワーク112を介して受信端末113に配信する。また、サーバ124は、必要に応じて、そのマルチメディアコンテンツと、それを再生するためのブラウザとからなるソフトウェアを配信する。なお、このソフトウェアは、上述したように、マルチメディアコンテンツ製作装置123より供給されたものとされてもよいし、サーバ124により、新たに生成されてもよい。

## 【0141】

サーバ124には、このような処理を実行するためのサーバアプリケーションソフトウェアがインストールされている。ネットワークやプロトコルの種類によりサーバアプリケーションの種類も異なるが、インターネット上でHTTPプロトコルを利用して配信を行う場合は、一般的なHTTPサーバが、サーバ124に相当する。

## 【0142】

ネットワーク112は、特に限定されないが、インターネット等のマルチメディアコンテンツを伝送可能な情報ネットワークである。

## 【0143】

受信端末113は、マルチメディアコンテンツを利用するユーザにより操作される。ユーザとして、一般的には、家庭における一般ユーザが想定されるが、それに限定されない。また、受信端末113は、マルチメディアコンテンツを利用可能なものであれば限定されず、パーソナルコンピュータの他、各種デジタル家

電製品でもよい。

【 0 1 4 4 】

ただし、本実施の形態が適用されるためには、受信端末 1 1 3 は、ネットワークを介して配信されるJava(R)アプリケーションソフトウェアとしてのマルチメディアコンテンツおよびブラウザ 1 3 1、および、配信されたJava(R)アプリケーションソフトウェアを実行するためのソフトウェア環境であるJava(R)実行環境 1 3 2、および、Java(R)実行環境 1 3 2をサポートするプラットフォーム 1 3 3を有する必要がある。

【 0 1 4 5 】

Java(R)実行環境 1 3 2は、この例においては、サン・マイクロシステムズ社が配布しているJRE(Java(R) Runtime Environment)とされるが、後述するように、他のJava(R)ソフトウェアとされてもよい。

【 0 1 4 6 】

次に、マルチメディアコンテンツ製作装置 1 2 3の構成例を、図 1 0を参照して説明する。

【 0 1 4 7 】

CPU 1 4 1は、ROM 1 4 2に記憶されているプログラム、または記憶部 1 4 8からRAM 1 4 3にロードされたプログラムに従って各種の処理を実行する。

【 0 1 4 8 】

RAM 1 4 3にはまた、CPU 1 4 1が各種の処理を実行する上において必要なデータなども適宜記憶される。

【 0 1 4 9 】

CPU 1 4 1、ROM 1 4 2、およびRAM 1 4 3は、バス 1 4 4を介して相互に接続されている。このバス 1 4 4にはまた、入出力インタフェース 1 4 5も接続されている。

【 0 1 5 0 】

入出力インタフェース 1 4 5には、キーボードなどよりなる入力部 1 4 6、画像（動画および静止画）を表示するディスプレイや音声を出力するスピーカなどよりなる出力部 1 4 7、ハードディスクなどより構成される記憶部 1 4 8、図 9

に示される、モノメディア製作装置 1 2 1、コンポーネント開発装置 1 2 2、および、サーバ 1 2 4 等と相互に通信するための通信処理を実行する通信部 1 4 9 が接続されている。

#### 【 0 1 5 1 】

入出力インタフェース 1 4 5 にはまた、必要に応じてドライブ 1 5 0 が接続され、磁気ディスク、光ディスク、光磁気ディスク、或いは半導体メモリなどのリムーバブル記録媒体 1 5 1 が適宜装着され、それらから読み出されたコンピュータプログラムが、必要に応じて記憶部 1 4 8 にインストールされる。

#### 【 0 1 5 2 】

なお、リムーバブル記録媒体 1 5 1 には、必要に応じて、モノメディア製作装置 1 2 1 により製作されたモノメディア、コンポーネント開発装置 1 2 2 により開発されたコンポーネント、または、マルチメディアコンテンツ製作装置 1 2 3 により製作されたマルチメディアコンテンツ等が記録される。

#### 【 0 1 5 3 】

図 1 1 は、サーバ 1 2 4 の構成例を表している。CPU 1 6 1 乃至リムーバブル記録媒体 1 7 1 は、上述した図 1 0 のマルチメディアコンテンツ製作装置 1 2 3 の CPU 1 4 1 乃至リムーバブル記録媒体 1 4 1 のうちの対応する名称のものと基本的に同様の構成とされているため、その説明は省略する。ただし、通信部 1 6 9 は、マルチメディアコンテンツ製作装置 1 2 3、および、ネットワーク 1 1 2 (図 9) と通信する処理を実行する。

#### 【 0 1 5 4 】

図 1 2 は、受信端末 1 1 3 の構成例を表している。CPU 1 8 1 乃至リムーバブル記録媒体 1 9 1 は、上述した図 1 0 のマルチメディアコンテンツ製作装置 1 2 3 の CPU 1 4 1 乃至リムーバブル記録媒体 1 4 1 のうちの対応する名称のものと基本的に同様の構成とされているため、その説明は省略する。ただし、通信部 1 6 9 は、ネットワーク 1 1 2 (図 9) と通信する処理を実行する。また、受信端末 1 1 3 は、図 1 2 の例では、パーソナルコンピュータの形態を取っているが、その形態はそれに限定されず、上述したように、デジタル家電製品等様々な形態を取ることが可能である。従って、入力部 1 8 6 および出力部 1 8 7 等も、その

受信端末 1 1 3 の形態により様々な形態を取ることが可能である。

【 0 1 5 5 】

また、図示はしないが、モノメディア製作装置 1 2 1、および、コンポーネント開発装置 1 2 2 のそれぞれも、上述した図 1 0 のマルチメディアコンテンツ製作装置 1 2 3 と基本的に同様の構成とされている。

【 0 1 5 6 】

次に、図 1 3 を参照して、本実施の形態が適用されるマルチメディアコンテンツ再生例の仕組みを説明する。

【 0 1 5 7 】

この例においては、クロスプラットフォームでのアプリケーション配信が可能である Java(R) の特性を生かし、コンテンツのみならずブラウザもネットワーク 1 1 2 を介して配信される。

【 0 1 5 8 】

図 1 3 の矢印 2 4 1 に示されるように、受信端末 1 1 3 に URL が入力されると、Java(R) 実行環境（厳密には後述するアプリケーションマネージャ） 1 3 2 は、矢印 2 4 2 に示されるように、URL で指定された、ネットワーク 1 1 2 上のサーバ 1 2 4 に対してアクセス要求を行い、サーバ 1 2 4 にアクセスする。そして、Java(R) 実行環境（厳密には後述するアプリケーションマネージャ） 1 3 2 は、ユーザにより指定されたマルチメディアコンテンツ 2 5 2、および、それを再生するためのブラウザ 2 5 1 からなる配信ソフトウェア（Java(R) アプリケーションソフトウェア） 2 4 3 を取得し、取得したブラウザ 2 5 1 を起動し、マルチメディアコンテンツ 2 4 3 のコンテンツ再生 2 4 4 を行う。例えば、受信端末 1 1 3 のディスプレイ（図 1 2 の出力部 1 8 7）には、コンテンツ再生 2 4 4 の結果として、後述する図 2 6 に示されるような画像 4 6 1 が表示される。

【 0 1 5 9 】

このように、ブラウザ 2 5 1 もネットワーク 1 1 2 を介して配信されることにより、ブラウザ 2 5 1 のアップデートをユーザが意識する必要がなくなる。また、Java(R) 実行環境 1 3 2 のみが標準化されれば、各種のアーキテクチャを持つブラウザとコンテンツを組み合わせたソフトウェア配信も自由に（標準化する必

要なく) 行うことが出来る。

【0160】

また、従来のHTMLブラウザのように、ブラウザ251を受信端末113に固定して実装し、運用することも可能である。この場合、ブラウザ251を、マルチメディアコンテンツ252を配信する毎に配信する必要がないので、配信データのデータサイズの抑制が可能になる。

【0161】

次に、図面を参照して、本実施の形態の動作環境について説明する。図14は、受信端末113側のソフトウェアスタックを表している。

【0162】

図14に示されるように、配信ソフトウェア243としては、Java(R)アプリケーション243-1が存在し、端末ソフトウェア261としては、アプリケーションマネージャ271、Java(R)実行環境132、および、プラットフォーム133が存在する。

【0163】

上述したように、Java(R)アプリケーション243-1は、マルチメディアコンテンツ252、および、ブラウザ251から構成される。マルチメディアコンテンツ252は、例えば、上述したシーンデータ191が複数個集められて構成される。

【0164】

アプリケーションマネージャ271は、図13に示されるように、ユーザからのURLの入力に従って、ネットワーク112上から、マルチメディアコンテンツ252およびブラウザ251(配信ソフトウェア243)をダウンロードし、実行するためのソフトウェアである。

【0165】

上述したJava(R)実行環境132は、例えば、拡張API(Application Program Interface)272、および、JRE(Java(R) Runtime Environment)273等から構成される。

【0166】

JRE 2 7 3 は、サン・マイクロシステムズ社が配布しているJava(R)アプリケーションの実行環境であり、主に、JVM(Java(R) Virtual Machine) 2 8 4 とJ2SE API(Java(R) 2 Platform Standard Edition API) 2 8 3 とから構成される。JVM 2 8 4 は、中間コードで表現されたJava(R)アプリケーションを解釈、および実行するためのソフトウェア (Java(R)バーチャルマシン) であり、J2SE API 2 8 3 は、Java(R)アプリケーション用に標準化されたクラスライブラリ群である。

## 【 0 1 6 7 】

拡張API 2 7 2 は、この例において、J2SEに追加されたJava(R)のクラスライブラリ群であり、主なものにJava(R)3D 2 8 2 とECMA(European Computer Manufacturers Association)スクリプトインタプリタ (商標) 2 8 1 がある。

## 【 0 1 6 8 】

Java(R)3D 2 8 2 は、サン・マイクロシステムズ社によって開発された、Java(R)プログラムで3Dを利用したアプリケーションを実現するためのクラスライブラリである。VRML (Virtual Reality Modeling Language) のような宣言的に3D仮想空間を構築するスクリプト言語と違い、プログラム中でメソッドを直接呼び出すことで3D仮想空間を構築することが可能になる。

## 【 0 1 6 9 】

ECMAスクリプトインタプリタ 2 8 1 は、この例においては、コンポーネントを操作するためのスクリプト言語としてECMAスクリプトが利用されるため、追加されたものである。ECMAスクリプトは、ネットスケープ・コミュニケーションズ社が開発したJava(R)スクリプトを標準化団体ECMAが標準化したものである。ECMAスクリプトは、一般のプログラミング言語のように条件分岐、および、繰り返しといった制御構造を有するので、柔軟なコンポーネント制御が可能である。

## 【 0 1 7 0 】

プラットフォーム 1 3 3 は、OS、ドライバ、および、ミドルウェア等からなるOS/ドライバ/ミドルウェア 2 7 3、並びに、ハードウェア (図 1 2 のCPU 1 8 1 乃至ドライブ 1 9 0 等) から構成される。

## 【 0 1 7 1 】

次に、交換フォーマットについて説明する。

## 【0172】

コンポーネントの実体は、そのコンポーネントを構成するJava(R)のクラスファイルと、そのクラスがインスタンス化され、必要なフィールドの値が設定された、すなわちカスタマイズされたオブジェクトである。

## 【0173】

シーンを構成する各コンポーネントは、Java(R).ioパッケージが提供するシリアライゼーション機構が利用され、シーン毎に1つのファイル（以下、インスタンスファイルと称する）に書き込まれる。この例においては、インスタンスファイルの拡張子は、“dat”とされる。ブラウザは、シーン再生時に、クラスファイル群とインスタンスファイルから各コンポーネントのオブジェクトを再構成する。

## 【0174】

図15は、ネットワーク112を介して受信端末113に実際に配信されるソフトウェアの構成例を表している。

## 【0175】

図15に示されるように、各シーンを構成するシーンデータ191のアーカイブファイル292は、インスタンスファイル301、コンポーネントのクラスファイル302、ECMASクリプトのコードがテキスト形式で記述されているスクリプトファイル303、および、テキストまたは静止画等のファイル304から構成される。

## 【0176】

このシーンデータ191のアーカイブファイル292の、インスタンスファイル301、クラスコンポーネントのクラスファイル302、および、スクリプトファイル303は、Java(R)が提供するJar形式という方式でアーカイブされる。コンポーネントが実行される際に利用されるテキストまたは静止画等のファイル（モノメディアファイル）群304も必要に応じてこのアーカイブに含められる。

## 【0177】

以下、シーンデータのアーカイブファイル群を構成するファイルを、シーンプ



ファイルと称する。このとき、最初に再生するシーンファイルの名称は"startup.jar"とされ、他のシーンファイルとは区別される。なお、"jar"はJar形式のアーカイブファイルの拡張子である。

【0178】

各シーンのインスタンスファイル301の名称は、シーンファイル292の名称と同じ名称とされる。従って、startupシーンのインスタンスファイルは"startup.dat"とされる。

【0179】

また、ブラウザ251もJar形式を利用してアーカイブされる。従って、配信ソフトウェアは、ブラウザ251のアーカイブファイル291と、各シーンに対応するシーンファイル（Jarファイル）292の群から構成される。

【0180】

Jarファイルにはマニフェストファイルと呼ばれるファイルが同梱される。これには、Jarファイルの各種属性の設定がテキストフォーマットで記されている。各属性はアトリビュートと呼ばれる。この例においては、startup.jarファイルのアトリビュートを拡張し、ブラウザのURL及びstartup以外の各シーンのURLが設定可能とされている。

【0181】

また、この例においては、各コンポーネントのインスタンスがファイルに書き込まれる場合、または、読み出される場合、Java(R).ioパッケージが提供するシリアライゼーション機構が利用されるが、これに限定されず、他にもXMLのようなテキスト形式の技術を利用することも可能である。

【0182】

次に、ブラウザ251とコンポーネントのインタフェースについて説明する。

【0183】

この例においては、ブラウザ251とコンポーネントのインタフェースは、Java(R)Beansの上に構築される。Java(R)Beansは、Java(R)が提供しているコンポーネントフレームワークである。

【0184】

個々のコンポーネントは、このJava(R)Beanの仕様を満たした上で、その機能に応じてmmcomponentパッケージで定められているコンポーネントインタフェースを実装しなければならない。

【 0 1 8 5 】

各インタフェースは、コンポーネントがブラウザから各種のサービスを受けるためのものであり、図 1 6 に示されるような継承関係を持っている。図中、下位のものは、上位のものを継承する。

【 0 1 8 6 】

そこで、図 1 6 を参照して、mmcomponentパッケージの詳細を説明する。

【 0 1 8 7 】

MComponentインタフェース（以下、単にMComponentと記述する）3 1 7 は、全てのコンポーネントが実装するべきインタフェースである。MComponent 3 1 7 は、getId()メソッドを提供する。getId()メソッドは、そのコンポーネントのID文字列を返し、ブラウザはこのメソッドにより得られるID値によって各コンポーネントを一意に識別する。また、MComponent 3 1 7 は、initialize()メソッドおよびdestroy()メソッドを提供する。コンポーネントは、initialize()メソッドで初期化の処理を、destroy()メソッドで終了の処理を、それぞれ行う必要がある。

【 0 1 8 8 】

MPresentableインタフェース（以下、単にMPresentableと記述する）3 1 6 は、何らかのプレゼンテーションを持つコンポーネントの動作を定義するものである。MPresentable 3 1 6 は、run()メソッドおよびstop()メソッドを提供する。run()メソッドによりプレゼンテーションが開始され、stop()メソッドによりプレゼンテーションが終了する。またプレゼンテーションが実行中か否かはgetActive()メソッドが返す真偽値により判定される。また、getInitiallyActive()メソッドは、コンポーネントがシーン開始時すぐにプレゼンテーションを開始されるべきか否かを真偽値によって返す。

【 0 1 8 9 】

MVisibleインタフェース（以下、単にMVisibleと記述する）3 1 5 は、可視な

コンポーネントの振る舞いを定義したものである。

#### 【 0 1 9 0 】

M3DComponent インタフェース（以下、単に M3DComponent と記述する） 3 1 3 は、3D のプレゼンテーションを持つコンポーネントの振る舞いを定義したものである。M3DComponent 3 1 3 は、getSubGraph() メソッドを提供する。このメソッドは Java(R) 3D API の BranchGroup クラスのインスタンスを返す。このオブジェクトは、そのコンポーネントの 3D プレゼンテーションの実体であり、ブラウザはそのメソッドによって得られた BranchGroup オブジェクトを、Java(R) 3D が提供するシーングラフに追加する。これにより、そのコンポーネントの 3D プレゼンテーションが可視状態になる。また、M3DComponent 3 1 3 は、get/setPosition() メソッド、get/setRotation() メソッド、get/setScale() メソッド等のメソッドを提供する。これらのメソッドにより、動画 3 1 2 を構成する各シーン中のコンポーネントのそれぞれの位置、姿勢、縮尺が指定されるので、提示の一意性が保証される。

#### 【 0 1 9 1 】

MBackground インタフェース（以下、MBackground と記述する） 3 1 4 は、3D 空間の背景を表現するコンポーネントの振る舞いを定義したものである。MBackground 3 1 4 は、getBackgroundGraph() メソッドを提供する。このメソッドは BranchGroup オブジェクトを返す。このオブジェクトは背景のプレゼンテーションの実体であり、ブラウザによってシーングラフに追加される。

#### 【 0 1 9 2 】

MInteractive インタフェース（以下、MInteractive と記述する） 3 1 8 は、ボタン 3 1 1 のようなインタラクティブなコンポーネントの振る舞いを定義したものである。MInteractive 3 1 8 を実装したコンポーネントは、フォーカス管理の対象となる。なお、MInteractive 3 1 8 を実装したコンポーネントの中で、任意の時点でフォーカスを保持しているものは、シーン中に高々 1 つ存在する程度である。

#### 【 0 1 9 3 】

ユーザが、キー群 9 1（図 8）を操作して、キー入力を行うと、ブラウザ 2 5

1 は、フォーカスを保持しているコンポーネント (MInteractive3 1 8 を実装したコンポーネント) の `keydown()` メソッド、`keyup()` メソッドを呼び出す。これにより、そのコンポーネントは、ユーザのキー入力を受け取ることが出来る。また、入力されたキーの種類が矢印キー 9 1 - 2 乃至 9 1 - 5 (図 8) である場合、フォーカスが遷移して、他のコンポーネントが新たにフォーカスを取得する。そのとき、それまでフォーカスを保持していたコンポーネントに対しては `blur()` メソッドが、新たにフォーカスを取得したコンポーネント (MInteractive3 1 8 を実装したコンポーネント) に対しては `focus()` メソッドが、それぞれ呼び出される。

## 【 0 1 9 4 】

そのコンポーネントがフォーカスを保持している時に、上下左右の矢印キー 9 1 - 2 乃至 9 1 - 5 による入力に対して、それぞれどのコンポーネントへフォーカスが遷移するかは、そのコンポーネントが ID 値として保持している。これらは MInteractive3 1 8 が提供する `getNavup()` メソッド、`getNavdown()` メソッド、`getNavLeft()` メソッド、および、`getNavright()` メソッドが返す値によって取得することが可能である。

## 【 0 1 9 5 】

図 1 7 を参照して、このようなフォーカス遷移の例を説明する。

## 【 0 1 9 6 】

いま、ID 値が「ボタン A」とされる MInteractive3 1 8 を実装しているコンポーネント (以下、ボタンと称する) 3 1 1 - 1、ID 値が「ボタン B」とされる MInteractive3 1 8 を実装しているコンポーネント (以下、単にボタンと称する) 3 1 1 - 2、ID 値が「ボタン C」とされる MInteractive3 1 8 を実装しているコンポーネント (以下、単にボタンと称する) 3 1 1 - 3、ID 値が「ボタン D」とされる MInteractive3 1 8 を実装しているコンポーネント (以下、単にボタンと称する) 3 1 1 - 4、および、ID 値が「ボタン E」とされる MInteractive3 1 8 を実装しているコンポーネント (以下、単にボタンと称する) 3 1 1 - 5 のそれぞれが、図 1 7 に示されるように配置されているものとする。

また、ボタン 3 1 1 - 1 の `getNavup()` メソッドは「ボタン B」を、`getNavdown()`

メソッドは「ボタンD」を、getNavLeft()メソッドは「ボタンE」を、getNavright()メソッドは「ボタンC」を、それぞれ返すものとする。

#### 【0197】

例えば、いま、ボタン311-1がフォーカスを保持していると仮定して、ユーザが上矢印キー91-2（図8）を押下した場合、ブラウザ251の後述するフォーカス管理部372（図23）は、ボタン311-1のgetNavup()メソッドを実行して、フォーカスの行き先がボタン311-2であると判断し、フォーカスをそのように遷移させ、ボタン311-1に対してblur()メソッドを、ボタン311-2に対してfocus()メソッドを実行する。同様に、下矢印キー91-4、左矢印キー91-5、または、右矢印キー91-3が押下された場合、対応するボタン311-4、ボタン311-5、または、ボタン311-3へフォーカスが遷移される。

#### 【0198】

図16に戻り、また、MInteractible318は、フォーカスの遷移が生じた際、即ち、focus()メソッドまたはblur()メソッドが呼び出された時にイベントを発生する。このイベントが発生されると、上述したECMASクリプトインタプリタ281（図14）は、MInteractible318が提供するgetOnfocus()メソッドまたはgetOnblur()メソッドが返す文字列をスクリプトとして解釈し、実行する。

#### 【0199】

MStreamインタフェース（以下、MStreamと記述する）319は、音声や動画312など、時間の経過に従い表現状態が変化するメディア（以下、ストリームと称する）の振る舞いを定義したものである。MStream319を実装したコンポーネント（以下、MStreamコンポーネントと称する）は同期処理の対象となる。そのインタフェースは、getStreamPosition()メソッド、または、setStreamPosition()メソッドを提供する。これによりストリームの再生位置の取得設定を行うことが可能になる。またgetStartLatency()メソッドによって、そのストリームの再生が可能になるまでの（準備に要する）待ち時間を取得することが可能になる。さらに、playStream()メソッド、または、stopStream()メソッドによってストリームの再生を行うことが可能になる。playStream()メソッドには、引数として

再生開始までの時間を指定することが可能である。

#### 【0200】

MStreamContainerインタフェース（以下、MStreamContainerと記述する）320は、複数のMStream319の同期制御を行うコンポーネントの振る舞いを定義したものである。MStreamContainer320は、同期を行う各MStreamコンポーネントの参照を保持する。

#### 【0201】

図18は、MStreamContainer320が、複数のMStreamコンポーネントを同期実行する際のシーケンス例を表している。

#### 【0202】

図18の例では、シーン中にMStreamContainer320を実装したコンテナ（以下、単にコンテナと記述する）と、MStreamコンポーネントとしての音声233および動画234が存在し、コンテナ332が音声233と動画234の同期制御を行うものとする。

#### 【0203】

矢印341に示されるように、まず、ECMAScriptインタプリタ281からコンテナ332に対して、playStream()メソッドの呼び出しが、引数0.0で実行される。コンテナ332は、即座に音声233と動画234をそれらのストリームの先頭から再生させる。

#### 【0204】

このとき、音声233と動画234とでストリームが再生されるまでの準備にかかる時間が異なることが想定される。この問題を解決するために、矢印342と矢印343に示されるように、コンテナ332は、音声233と動画234に対して、getStartLatency()メソッドを呼び出し、各コンポーネントがストリームを再生できるまでにかかる時間t1およびt2を取得する。

#### 【0205】

次に、コンテナ332は、これらの時間（t1, t2）の最大値t0を求め、矢印344と矢印345に示されるように、この値を引数として各コンポーネントのplayStream()メソッドを実行する。時間t0は、時間t1、または時間t2と等しいか

、それ以上の時間なので、時間 $t_0$ が経過するまでに、音声233および動画234のストリーム再生の準備は終了していると考えられ、 $t_0$ の時間が経過した後に、これらのコンポーネントはストリームを同期して再生し始めることが可能になる（音声の再生346と動画の再生347が同期して実行される）。

#### 【0206】

また、制御される側のコンポーネントが3つ以上ある場合も同様な処理によりストリームの同期再生が可能である。

#### 【0207】

図16に戻り、MStreamContainer320は、その同期機構の機能や実現方法まで定義するものではない。従って、上述した（図18に示される）ロジック以外のロジックによる同期制御コンポーネントの実現も可能である。なお、図18に示されるシーケンス例は、サン・マイクロシステムズ社によって開発されたJMF（Java(R) Media Framework）（商標）における同期実行処理とほぼ同じ考えに基づいたものである。

#### 【0208】

以上、mmcomponentパッケージを構成する各インタフェースに関して説明したが、図16の例では、これらのインタフェースを実装するコンポーネントの例として、ボタン311と動画312が示されている。ボタン311は、3D仮想空間上に実体を持つのでM3DComponent313を実装し、さらにリモコンキー91（図8）によるインタラクティブな操作が可能となるように、MInteractable318を実装している。また、動画312は、3D仮想空間上に実体を持つのでM3DComponent313を実装し、さらに時間軸を持つ動画データの再生を行うのでMStream319を実装している。

#### 【0209】

次に、図19を参照して、ブラウザのソフトウェアの構成例を説明する。

#### 【0210】

ブラウザ251は、ローダ部362、シーンデータ部363、描画部364、ECMAScript処理部365、イベント処理部366、および、制御部361から構成される。

## 【 0 2 1 1 】

ローダ部 3 6 2 のコンポーネントローダ 3 6 7 は、指定されたシーンのシーンファイル（アーカイブファイル） 2 9 2 （図 1 5）をロードし（入力し）、上述したような ID 値を基に管理されているコンポーネント群 3 6 8 と、スクリプトファイル 3 0 3 （図 1 5）に対応するスクリプトデータ 3 6 9 に分割し、シーンデータ部 3 6 3 に供給する。具体的には、コンポーネントローダ 3 6 7 は、ロードしたシーンファイル 2 9 2 （図 1 5）に含まれるクラスファイル 3 0 2 の群と、インスタンスファイル 3 0 1 から、シーンを構成するオブジェクトファイルを再構成する。

## 【 0 2 1 2 】

スクリプトデータ 3 6 9 は、スクリプトインタプリタ 3 7 1 に供給される。スクリプトインタプリタ 3 7 1 は、上述した ECMA スクリプトのインタプリタであり、スクリプトデータ 3 6 9 を走査し、コンテンツ実行時にそのデータの内容によってコンポーネントを操作したり、またコンポーネントが発生するイベントを補足して指定された関数を実行したりする。

## 【 0 2 1 3 】

ECMA スクリプト処理部 3 6 5 は、このような関数として、拡張関数 3 7 0 を有しており、対応する関数がスクリプトインタプリタ 3 7 1 に適宜提供される。即ち、拡張関数 3 7 0 は、シーン遷移等の機能を提供するための関数群であり、ECMA スクリプトから呼び出すことができる。

## 【 0 2 1 4 】

描画部 3 6 4 の 3D 描画エンジン 3 7 3 は、コンポーネント群 3 6 8 のうちの対応するコンポーネントを取得し、再生画面として受信端末 1 1 3 のディスプレイ（図 1 2 の出力部 1 8 7）に表示する 3D グラフィックスを描画する部分である。この例においては、3D 描画エンジン 3 7 3 は、Java(R) 3D の描画エンジンを利用するが、これに限定されるものはない。

## 【 0 2 1 5 】

イベント処理部 3 6 6 のイベント/フォーカス管理部 3 7 2 は、入力部 1 8 6 からの入力を、フォーカスを保持しているコンポーネントに配布する部分である



。また、上述したように、入力部 1 8 6 の入力、図 8 の上下左右キー 9 1 - 2 乃至 9 1 - 5 の入力の場合には、フォーカスの遷移を行う。

#### 【 0 2 1 6 】

制御部 3 6 1 は、ブラウザ 2 5 1 本体の起動、終了などの振る舞いを管理する部分である。

#### 【 0 2 1 7 】

次に、図 2 0 を参照して、ブラウザ 2 5 1 によるコンポーネントのライフサイクル管理について説明する。ブラウザ 2 5 1 は、シーンを読み込んだ後、シーンに含まれるコンポーネントのライフサイクルを管理するが、この場合のコンポーネントの状態は、図 2 0 に示されるようになる。

#### 【 0 2 1 8 】

即ち、コンポーネントの状態は、シリアライズされた状態（以下、状態 A と称する）、デシリアライズされた状態（以下、状態 B と称する）、初期化され、かつアクティブでない状態（以下、状態 C と称する）、アクティブな状態（以下、状態 D と称する）、または、破棄された状態（以下、状態 E と称する）のうちのいずれかの状態とされる。

#### 【 0 2 1 9 】

次に、この図 2 0、および、図 2 1 のフローチャートを参照して、ブラウザ 2 5 1 のシーン起動処理について、説明する。

#### 【 0 2 2 0 】

指定されたシーンを起動する際、図 2 1 のステップ S 1 において、ブラウザ 2 5 1 は、シーンファイル 2 9 2（図 1 5）のインスタンスファイル 3 0 1 中にシリアライズされたコンポーネント（図 2 0 の状態 A のコンポーネント）群をデシリアライズ（ロード）する。

#### 【 0 2 2 1 】

即ち、図 2 0 の矢印 3 9 1 に示されるように、readObject（）メソッドが実行されて、コンポーネントは状態 A から状態 B に遷移する。

#### 【 0 2 2 2 】

図 2 1 のステップ S 2 において、ブラウザ 2 5 1 は、ステップ S 1 の処理でロ

ードしたコンポーネントを、そのID値をキーとして自らに登録し、ステップS3において、そのコンポーネントのinitialize()メソッドを実行して、そのコンポーネントを初期化する。これにより、図20の矢印392に示されるように、コンポーネントは、状態Bから状態Cに遷移する。

#### 【0223】

図21のステップS4において、ブラウザ251は、ステップS3の処理で初期化したコンポーネントが、M3DComponent313（図16）またはMBackground314（図16）を実装しているか否かを判定し、それらを実装していると判定した場合、ステップS5において、そのコンポーネントを3D仮想空間へ追加し、ステップS6に進む。

#### 【0224】

これに対して、ブラウザ251は、ステップS4において、コンポーネントが、M3DComponent313またはMBackground314を実装していないと判定した場合、ステップS5の処理は実行せずに、ステップS6に進む。

#### 【0225】

ステップS6において、ブラウザ251は、ECMASクリプトからコンポーネントを操作したり、コンポーネントが発生するイベントをECMASクリプトに通知したりすることが可能となるように、コンポーネントを、スクリプトインタプリタ371（図19）へバインドする（コンポーネントへの参照をスクリプトインタプリタ371に渡す）。

#### 【0226】

コンポーネントがスクリプトインタプリタ371にバインドされていることで、スクリプトコードの中からそのコンポーネントのメソッドが実行されたり、コンポーネントがなんらかのイベントを発した際に、その種類に応じて予め指定されたスクリプトコードが実行されたり（このような処理を、以下、イベントハンドラの起動と称する）することが可能になる。

#### 【0227】

ステップS7において、ブラウザ251は、コンポーネントが、MPresentable316（図16）を実装しており、かつinitiallyActive()メソッドがtrueを返し

たか (initiallyActive=true) 否かを判定する。

【 0 2 2 8 】

ステップ S 7 において、ブラウザ 2 5 1 は、コンポーネントが、MPresentable 3 1 6 を実装しており、かつ、initiallyActive=trueであると判定した場合、ステップ S 8 において、そのコンポーネントに、run()メソッドを実行して、プレゼンテーションを開始する。

【 0 2 2 9 】

これにより、図 2 0 の矢印 3 9 3 に示されるように、コンポーネントは、状態 C から状態 D に遷移する。即ち、そのコンポーネントは、アクティブに設定される。

【 0 2 3 0 】

なお、そのコンテンツが再生されている最中に、矢印 3 9 4 に示されるように、スクリプトインタプリタ 3 7 1 (図 1 9) から、stop()メソッドが呼び出されると、そのコンポーネントは、状態 D (アクティブ状態) から、状態 C (非アクティブ状態) に遷移する。さらに、矢印 3 9 3 に示されるように、スクリプトインタプリタ 3 7 1 から、run()メソッドが呼び出されると、そのコンポーネントは、再び、状態 C から、状態 D に遷移する。

【 0 2 3 1 】

即ち、コンテンツの再生中は、そのコンポーネントは、適宜状態 C と状態 D の間で切り替わる。

【 0 2 3 2 】

図 2 1 に戻り、ブラウザ 2 5 1 は、ステップ S 8 の処理を実行した後、ステップ S 9 に進む。

【 0 2 3 3 】

一方、ステップ S 7 において、ブラウザ 2 5 1 は、コンポーネントが、MPresentable 3 1 6 を実装していないか、または、initiallyActive=trueではないと判定した場合、ステップ S 8 の処理を実行せずに、ステップ S 9 に進む。

【 0 2 3 4 】

ステップ S 9 において、ブラウザ 2 5 1 は、シーンファイル 2 9 2 からスクリ

プトファイル303を読み込み、それを解釈し、実行する。

【0235】

次に、図20と、図22のフローチャートを参照して、ブラウザ251のシーン終了処理を説明する。

【0236】

コンテンツ終了時、図22のステップS21において、ブラウザ251は、各コンポーネントに、destroy()メソッドを実行し、終了処理を実行する。

【0237】

これにより、コンポーネントは、図20の矢印395および矢印396に示されるように、状態Cまたは状態Dから、状態Eに遷移する。

【0238】

ステップS22において、ブラウザ251は、そのコンポーネントの登録を削除する。

【0239】

ステップS23において、ブラウザ251は、コンポーネントが、M3DComponent313（図16）またはMBackground314（図16）を実装しているか否かを判定し、コンポーネントが、M3DComponent313またはMBackground314を実装していると判定した場合（コンポーネントが3D仮想空間中に表示されている場合）、ステップS24において、コンポーネントを3D仮想空間から削除し、ステップS25に進む。

【0240】

これに対して、ステップS23において、ブラウザ251は、コンポーネントが、M3DComponent313またはMBackground314を実装していないと判定した場合（コンポーネントが3D仮想空間中に表示されていない場合）、ステップS24の処理を実行せずに、ステップS25に進む。

【0241】

ステップS25において、ブラウザ251は、そのコンポーネントの、スクリプトインタプリタ371とのインタプリタバインドを解放する。

【0242】

次に、図23を参照して、ブラウザ251のイベント配布及びフォーカス管理の処理例について説明する。

#### 【0243】

図23は、ユーザからのキー入力が発生した際の、イベント/フォーカス管理部372（図19）、MInteractible318（図16）を実装した2つのコンポーネント（図17のボタン311-1およびボタン311-2に相当）、および、スクリプトインタプリタ371（図19）間のメッセージ交換のシーケンスを表している。

#### 【0244】

図23の矢印411に示されるように、ユーザ401が、任意のキー91（図8）を押下すると、イベント/フォーカス管理部372は、矢印412に示されるように、現在フォーカスを保持しているコンポーネント（ここでは、ボタン311-1とされる）に対して、keypress()メソッドを実行することでその通知を行う。

#### 【0245】

次に、入力されたキーが矢印キー91-2乃至91-5のうちのいずれかの場合には、フォーカスの遷移が行われる。

#### 【0246】

いま、ユーザ401は、上矢印キー91-2を押下したものとする。まず、イベント/フォーカス管理部372は、矢印413に示されるように、ボタン311-1に対してgetNavup()メソッドを実行し、矢印414に示されるように、その返回值よりフォーカス遷移先のコンポーネントのIDを取得する（ここではID値は、「ボタンB」とされる）。

#### 【0247】

次に、イベント/フォーカス管理部372は、矢印415に示されるように、ボタン311-1に対してblur()メソッドを実行し、フォーカスを喪失したことを通知する。

#### 【0248】

ボタン311-1は、フォーカス喪失に伴ってイベント（onbluerイベント）

を発生する。スクリプトインタプリタ 3 7 2 は、矢印 4 1 6 に示されるように、この onblur イベントを取得し、矢印 4 1 7 に示されるように、ボタン 3 1 1 - 1 が getOnblur () によって返す文字列をスクリプトとして解釈し、実行する（イベントハンドラを実行する）。

#### 【 0 2 4 9 】

次にイベント/フォーカス管理部 3 7 2 は、矢印 4 1 8 に示されるように、フォーカス遷移先のコンポーネントボタン 3 1 1 - 2 に対して、focus () メソッドを実行し、その旨を通知する。

#### 【 0 2 5 0 】

ボタン 3 1 1 - 2 は、フォーカス取得に伴ってイベントを発生する。スクリプトインタプリタ 3 7 1 は、矢印 4 1 9 に示されるように、この onfocus イベントを取得し、矢印 4 2 0 に示されるように、ボタン 3 1 1 - 2 が getOnfocus () によって返す文字列をスクリプトとして解釈実行する（イベントハンドラを実行する）。

#### 【 0 2 5 1 】

その後、ユーザ 4 0 1 が、矢印 4 2 1 に示されるように、押下していたキーからその指を離すと（キー入力を実行すると）、イベント/フォーカス管理部 3 7 2 は、矢印 4 2 2 に示されるように、フォーカスを保持しているコンポーネントボタン 3 1 1 - 2 に対して keyup () メソッドを実行することでその通知を行う。

#### 【 0 2 5 2 】

次に、本実施の形態が適用されるアプリケーションマネージャの説明をする。

#### 【 0 2 5 3 】

アプリケーションマネージャ 2 7 1（図 1 4）は、図 1 3 に示されるように、ユーザによる URL の入力（矢印 2 4 1）に従って、ネットワーク 1 1 2 上から、マルチメディアコンテンツ 2 5 2 およびブラウザ 2 5 1（配信ソフトウェア 2 4 3）をダウンロードし、実行するためのソフトウェアであり、ブラウザ 2 5 1 の起動後は、制御はブラウザ 2 5 1 側に渡される。

#### 【 0 2 5 4 】

図 2 4 のフローチャートを参照して、アプリケーションマネージャの処理例に

ついて説明する。

【0255】

いま、図13に示されるように、ユーザにより所定のコンテンツ（startupシーン）のURLが受信端末113に供給された場合（矢印241に示されるように入力された場合）、アプリケーションマネージャ271は、ステップS41において、そのstartupシーンのURLを入力する。

【0256】

ネットワーク112がインターネットであり、インターネット上でHTTPプロトコルが利用される場合、そのURLは、例えば、次のように記述される。

URL「jar:http://w.w.w.sony.co.jp/.../startup.jar!/startup.dat」

【0257】

このURLは、指定されたURL上のシーンファイル（アーカイブファイル（startup.jar）292（図15）中の、インスタンスファイル（startup.dat）301を指している。

【0258】

図24に戻り、ステップS42において、アプリケーションマネージャ271は、ステップS41の処理で入力したURL上のstartupシーンファイル292を取得する（ダウンロードする）。

【0259】

ステップS43において、アプリケーションマネージャ271は、そのstartupシーンファイル292内に指定されているstartup以外の各シーンのURLを基に、それらのURLからクラスファイルをロードできるように、パスを設定する。

【0260】

ステップS44において、アプリケーションマネージャ271は、そのstartupシーンファイル292内に指定されているブラウザ251のURLを基に、それを取得するためのパスを設定する。

【0261】

ステップS45において、アプリケーションマネージャ271は、ブラウザ251のアーカイブファイル291をロードし（取得し）、ブラウザ251を起動

する。

【0262】

その後、アプリケーションマネージャ271は、その処理をブラウザ251に渡し、ブラウザ251は、ステップS46において、そのstartupシーンを再生する。

【0263】

次に、図25を参照して、本実施の形態が適用されるオーサリングシステム441について説明する。

【0264】

オーサリングシステム441は、基本的に、上述したように（図9に示されるように）、マルチメディアコンテンツ製作装置123にインストールされるソフトウェアである。

【0265】

図25に示されるように、オーサリングシステム441は、ウィンドウベースのユーザフレンドリーなGUIを有している。オーサリングシステム441は、上述したマルチメディアコンテンツを製作可能なものであれば、特に限定されないが、この例においては、以下に示す（41）乃至（45）の機能を有するものとする。

【0266】

（41） マルチメディアコンテンツを新規に作成する作成機能、編集したマルチメディアコンテンツを保存する保存機能、および、そのマルチメディアコンテンツを実行する実行機能

保存機能としては、マルチメディアコンテンツを実行可能な状態で保存する機能、または、オーサリングシステム441独自の内部形式で保存する機能が考えられる。

【0267】

（42） 本実施の形態のアーキテクチャに基づいて開発されたコンポーネントを読み込み、ユーザに提示するコンテンツ提示機能

（43） マルチメディアコンテンツに対して、任意のコンポーネントを追加



したり、またそれらのコンポーネント群のカスタマイズを行ったりするための、  
ビジュアルなユーザーインタフェース環境を提供するユーザーインタフェース環境機能

(44) マルチメディアコンテンツの画面構成情報や、マルチメディアコンテンツに含まれるコンポーネント群の情報をビジュアルに提示する情報提示機能

(45) ECMAスクリプトの編集を支援するスクリプト編集支援機能

【0268】

また、図25に示されるように、オーサリングシステム441は、ディスプレイ上に、以下に示す項目(51)乃至(54)を表示させる。

【0269】

(51) コンポーネントリスト451

Java(R) Beansは、Jarアーカイブ形式で配布することが可能である。コンポーネントリスト451は、このような形態で提供されるコンポーネントをロードし、リスト形式で表示する。このようなコンポーネント群は、オーサリングシステム441内の特定のディレクトリに配置される。また、コンポーネントリスト451は、ユーザがリスト中で選択したコンポーネントを製作中のコンテンツに追加する。

【0270】

(52) コンテンツツリー452

コンテンツツリー452は、マルチメディアコンテンツのデータ構造をツリー形式で表示する。また、コンテンツツリー452は、マルチメディアコンテンツのデータ構造が変更された場合(シーンの追加若しくは削除、または、コンポーネントの追加若しくは削除が行われた場合)には、それをツリーに反映させる。コンテンツツリー452は、ユーザがツリー中のノードを選択した場合に、それに対応したシーンまたはコンポーネントを選択状態にする。

【0271】

(53) プロパティテーブル454

プロパティテーブル454は、選択中のコンポーネントのカスタマイズ情報をテーブル形式で表示する。これは、テーブル中で選択した属性の値の編集を可能

にする。また、プロパティテーブル454は、他の処理によってそのコンポーネントの属性値が変更された場合も、それをテーブルに反映する。

#### 【0272】

##### (54) レイアウトパネル455

レイアウトパネル455は、選択中のシーンに対応した3D仮想空間を提示する。レイアウトパネル455は、そのシーンにコンポーネントが追加された場合に、それを表示する。レイアウトパネル455は、そのシーンのデータ構造が変更された場合（シーンの追加若しくは削除、または、コンポーネントの追加若しくは削除が行われた場合）、それを反映する。これにより、コンテンツの製作者は、編集中のコンテンツの状態を即座に確認することが出来る。また、レイアウトパネル455は、マウスによるコンポーネントの位置、姿勢の設定が可能である。

#### 【0273】

##### (55) スクリプトエディタ453

スクリプトエディタ453は、選択中のシーンの、ECMASクリプトのコードを入力、編集するためのエディタである。

#### 【0274】

##### (56) その他

その他、オーサリングシステム441は、メニュー、および、ツールバー等を備える。

#### 【0275】

なお、オーサリングシステム441は、図25に示されるようなグラフィカルな環境を持たないコンテンツ製作環境でもよい。

#### 【0276】

上述したように、このようなオーサリングシステム441により製作されたマルチメディアコンテンツは、図9に示されるように、サーバ124に提供され、図13に示されるように、上述したブラウザ251とともに1つの配信ソフトウェア243として、サーバ124よりネットワーク112を介して受信端末113に配信される。なお、上述したように、マルチメディアコンテンツとブラウザ

251を組み合わせて配信ソフトウェア243にする処理は、オーサリングシステム441が実装されているマルチメディアコンテンツ製作装置123（図9）により行われてもよいし、サーバ124により行われてもよい。

【0277】

受信端末113は、配信ソフトウェア243に含まれるブラウザ251を利用して、配信ソフトウェア243に含まれるマルチメディアコンテンツ252を、そのディスプレイ等に表示させる（コンテンツ再生244の状態にさせる）。このとき、ディスプレイには、図26に示されるような画像461がブラウザ上に表示される。

【0278】

このように、オーサリングシステム441は、コンテンツ製作者（マルチメディアコンテンツ製作装置123（図9）のユーザ）に対して、効率的な開発環境を提供し、また、ソフトウェアに関する高度な知識を要しないコンテンツ製作者でも容易に利用することが可能である。従って、特に、デザイナーやクリエイターといった職種のコンテンツ製作者は、オーサリングシステム441を利用することで、品質の高いコンテンツを容易に製作することが可能になる。

【0279】

しかしながら、一般的に、オーサリングシステム441のシステムは、その開発にコストがかかるので、ユーザ（コンテンツ製作者）は、そのようなシステムを安価に入手することは困難である。

【0280】

また、上述したような本実施形態が適用される技術を広く普及させるためには、評価または研究することを目的とするマルチメディアコンテンツの製作が必要になると思われる。

【0281】

従って、ユーザ（コンテンツ製作者）に対して、オーサリングシステム441のようなシステムを入手しなくても、マルチメディアコンテンツを手軽、または安価に製作できる環境を提供する必要がある。

【0282】

そこで、本実施形態においては、オーサリングシステム441の代わりとなるソフトウェアとして、テキストによりデータ構造を表す技術（以下、テキストデータ構造表現技術と称する）を利用するフォーマットで記述されたマルチメディアコンテンツのフォーマットを、上述したようなサーバ124（図9と図13）がそのマルチメディアコンテンツを配信する場合に利用するフォーマット（上述した図15に示されるようなフォーマット）に変換する変換ソフトウェアも用意されている。

【0283】

なお、テキストデータ構造表現技術の種類は、特に限定されず、上述した変換ソフトウェアの開発者が独自に開発したものを利用することも可能であるし、XML (Extensible Markup Language) 等のオープンな規格に準じた方式を利用することが可能である。

【0284】

この場合、変換ソフトウェアの開発者は、フォーマット変換の対象となるマルチメディアコンテンツ（変換前のマルチメディアコンテンツ）のフォーマットとして、自分自身が独自に開発したテキストデータ構造表現技術を採用する変換ソフトウェアを開発することで、その開発コストを抑えることが可能である。

【0285】

その開発コストをさらに抑えたい場合、変換ソフトウェアの開発者は、フォーマット変換の対象となるマルチメディアコンテンツのフォーマットとして、オープンな規格に準じたテキストデータ構造表現技術を採用する変換ソフトウェアを開発することで、それが可能になる。

【0286】

その結果、ユーザ（コンテンツ製作者）は、そのような変換ソフトウェアを容易に入手することが可能になり、上述したオーサリングシステム441を入手しなくても、その変換ソフトウェアを利用して、オーサリングシステム441が製作するものと同じのフォーマットのマルチメディアコンテンツを製作することが可能になる。

【0287】

従って、この例の変換ソフトウェアは、例えば、XMLを利用するテキストフォーマット（以下、XMLフォーマットと称する）で記述されたマルチメディアコンテンツ（以下、このようにXMLフォーマットで記述されたマルチメディアコンテンツ（データ）を、XMLデータと称する）のフォーマットを、上述した図15に示されるようなフォーマットに変換するソフトウェアとされる。

## 【0288】

XMLは、上述したW3Cが規格化を行った、テキストを利用する汎用的なデータ記述言語である。XMLにおいては、HTMLと同様に、マークアップによりデータ構造が表現される。

## 【0289】

テキストデータ構造表現技術として、XMLが採用される場合の利点として、次の5点が挙げられる。

## 【0290】

即ち、第1点目として、XMLは、他の汎用的なテキストデータ構造表現技術と比較して、極めて普及度が高い点である。第2点目として、XMLの規格は、オープンになっており利用しやすい点である。第3点目として、XMLの文法は、シンプルで理解しやすい点である。第4点目として、XMLで表現されたデータを処理するためのライブラリが、数多く存在する点である。第5点目として、XMLを利用したデータの製作を支援するアプリケーションソフトウェア（専用エディタ等）が数多く存在し、無償または比較的安価に入手できる場合が多い点である。

## 【0291】

図27は、そのような本実施形態が適用される変換ソフトウェアを実装したマルチメディアコンテンツ製作装置123のソフトウェアの構成例を表している。

## 【0292】

図27に示されるように、マルチメディアコンテンツ製作装置123には、テキストエディタ501、変換ソフトウェア502、および、DTD（Document Type Definition）503が設けられている。

## 【0293】

テキストエディタ501は、コンテンツ製作者517が、XMLフォーマットに

よりマルチメディアコンテンツを記述する場合に利用するアプリケーションソフトウェアである。即ち、テキストエディタ501は、XMLデータ（XMLフォーマットのマルチメディアコンテンツ）511を生成する。この場合、図27の例では、各シーンのECMASクリプトのソースコード512もXMLデータ511の一部として記述される。なお、スクリプトのソースコード512は、XMLデータ511とは異なる別のファイルとして出力されてもよい。

【0294】

テキストエディタ501は、XMLデータ511を生成可能なものであれば限定されない。例えば、テキストエディタ501として、iPAD（アイパッド）（商標）、Athens Editor（商標）等の比較的安価に入手できるXML汎用エディタが利用可能である。

【0295】

変換ソフトウェア502は、XMLデータ511、テキスト/静止画等のファイル513、および、コンポーネントのクラスファイル514から、上述したようなサーバ124（図9と図13）がそのマルチメディアコンテンツを配信する場合に利用するフォーマットのマルチメディアコンテンツ516（図15に示されるものと同様の複数のシーンファイル292からなるマルチメディアコンテンツ516）に変換するソフトウェアである。

【0296】

なお、テキスト/静止画等のファイル513は、図15のテキスト/静止画等のファイル304に対応するものであり、図27の例では、XMLデータ511に組み込むものとしてモノメディア製作装置121（図9）よりマルチメディアコンテンツ製作装置123に予め供給されている。

【0297】

また、コンポーネントのクラスファイル514は、図15のコンポーネントのクラスファイル302に対応するものであり、図27の例では、XMLデータ511に含まれるコンポーネントのクラスファイルとしてコンポーネント開発装置122（図9）よりマルチメディアコンテンツ製作装置123に予め供給されている。

## 【0298】

図27の例では、さらに、変換ソフトウェア502は、マルチメディアコンテンツ516と、上述したブラウザのアーカイブファイル291を組み合わせて配信ソフトウェア515とし、それをサーバ124に供給する処理も実行する。

## 【0299】

ただし、上述したように、この処理の実行場所は、特に限定されず、変換ソフトウェア502とは異なるソフトウェア（図示せず）により実行されてもよいし、マルチメディアコンテンツ製作装置123ではなくサーバ124（図9）により実行されてもよい。

## 【0300】

DTD503は、XMLフォーマットにより記述されたデータ（この例では、XMLデータ511）の中に現れるタグの種類と出現規則を定義したものがデータ化されたものである。DTD503の詳細、および、タグについては、後述する。

## 【0301】

次に、図28のフローチャートを参照して、図27の変換ソフトウェア502の処理を説明する。

## 【0302】

なお、この例においては、XMLデータ511は、例えば、以下のようなXMLフォーマットのマルチメディアコンテンツとされる。

## 【0303】

即ち、この例のXMLデータ511が、変換ソフトウェア502の後述するステップS61乃至S71の処理により、マルチメディアコンテンツ516に変換され、ブラウザのアーカイブファイル291とともに配信ソフトウェア515として、サーバ124（図9）に供給され、さらに、サーバ124よりその配信ソフトウェア515がネットワーク112（図9）を介して受信端末113（図9）に配信された場合、受信端末113が、配信されたブラウザのアーカイブファイル291に対応するブラウザを利用して、配信されたマルチメディアコンテンツ516を再生させると、受信端末113のディスプレイ（図12の出力部187）には、図29に示されるような画像521が表示される。

## 【0304】

具体的には、この例のXMLデータ511（図29の画像521を表示させるXMLデータ511）は、図30乃至図61のように記述される。

## 【0305】

なお、以下、図30乃至図61に示されるXMLデータ511（部分511-1乃至511-32からなるXMLデータ511）のうちの、“<component>”や“<scene>”など、“<”と“>”で囲まれた部分を、HTMLと同様に、タグと称する。また、1組のタグ（例えば“<component>”と“</component>”）で挟まれた部分を、要素と称する。

## 【0306】

例えば、XMLデータ511のうちの図30に示される部分511-1は、次のことを表現している。

## 【0307】

即ち、XMLデータ511は、部分aに示されるように、“welcome”という名称のマルチメディアコンテンツであり、部分bに示されるように、“startup”という名称のシーンが含まれている。部分cに示されるように、シーン“startup”には、SSpotLightクラスを実体として持つコンポーネントが含まれている。また、部分dと部分eに示されるように、このコンポーネントの“id”属性の値は、文字列であり、“sSpotLight1”である。511-1のその他の部分、および、部分511-2乃至511-32の説明は、冗長となるので省略する。

## 【0308】

図28に戻り、はじめに、ステップS61において、図27の変換ソフトウェア502は、DTD503によって、XMLデータ511の検証を行う。

## 【0309】

即ち、XMLフォーマットで記述されたデータ（いまの場合、上述したXMLデータ511）の中に現れるタグの種類と出現規則に関しては、そのデータ構造毎に定義する必要がある。定義方法に関しては何種類かの方式が提案されているが、最も広く利用されているものが上述したDTDである。

## 【0310】



この例においては、上述したように、DTD 5 0 3 がマルチメディアコンテンツ製作装置 1 2 3（図 1 0 の記憶部 1 4 8 等）に予め記憶されており、変換ソフトウェア 5 0 2 は、実際に製作された XML データ 5 1 1 を、DTD 5 0 3 と照らし合わせて規則にしたがっているか否かをチェックする。この処理を、ここでは、検証と称する。

#### 【 0 3 1 1 】

この例においては、例えば、DTD 5 0 3 は、図 6 2 の部分 5 0 3 - 1 と図 6 3 の部分 5 0 3 - 2 ように記述される。

#### 【 0 3 1 2 】

なお、図 6 2 と図 6 3 に示される DTD 5 0 3 のうちの、“\*”は、「繰り返し」を表しており、例えば、図 6 2 の部分 A に示される `<!ELEMENT scene (script, component*)>` は、要素 “scene” の中には、要素 “script” が 1 回出現し、次に、要素 “component” が任意の回数出現することを表している。

#### 【 0 3 1 3 】

また、“#REQUIRED”は、「必須指定」を、“ID”は属性が取り得る値（識別子（ユニークな値））を、それぞれ表しており、例えば、図 6 2 の部分 B に示される `<!ATTLIST scene name ID #REQUIRED>` の部分は、要素 “scene” には属性 “name” があり、その値 (ID) は必ず指定する必要がある、かつ XML データ 5 1 1 中で一意でなければならないことを表している。即ち、同じ名前（属性 “name” の値 (ID)）のシーン（要素 “scene”）が複数存在してはいけないことを表している。

#### 【 0 3 1 4 】

また、図 6 2 の部分 A に示される `<!-- コンテンツを構成するシーン -->` は、`<!ELEMENT scene (script, component*)>` の注釈を表しており、図 6 2 の部分 B に示される `<!-- シーンの名称（必須指定） -->` は、`<!ATTLIST scene name ID #REQUIRED>` の注釈を表している。

#### 【 0 3 1 5 】

DTD 5 0 3 のその他の部分（図 6 2 の部分 A と部分 B を除く部分 5 0 3 - 1、および、図 6 3 の部分 5 0 3 - 2 に示される各タグ）も同様に、その部分（タグ）に対応する注釈（`<!--        -->` の部分）が記述されている。従って、その他の

部分についての説明は、冗長となるので省略するが、所定の部分（タグ）の意味は、その部分（タグ）に対応する注釈により容易に推定することが可能である。

【0316】

このようにして、図28のステップS61の処理でDTD503によって、XMLデータ511の検証が行われると、ステップS62の処理が実行される。

【0317】

即ち、XMLデータ（XMLフォーマットのマルチメディアコンテンツ）511は、複数のシーンから構成されているので、ステップS62において、変換ソフトウェア502は、それらの複数のシーンの中から、これから処理を行うシーン（以下、対象シーンと称する）を選定する。

【0318】

ステップS63において、変換ソフトウェア502は、対象シーンに含まれる各コンポーネントのそれぞれのクラスの種類を、XMLデータ511より取得する。

【0319】

ステップS64において、変換ソフトウェア502は、ステップS63の処理で取得されたクラスの種類に基づいて、各クラスのそれぞれに対応するコンポーネントのクラスファイル514をロードする（取得する）。

【0320】

ステップS65において、変換ソフトウェア502は、対象シーンに含まれる各コンポーネントのそれぞれに相当するJava(R)オブジェクトを生成する。

【0321】

ステップS66において、変換ソフトウェア502は、XMLデータ511で指定されている属性について、その値を、ステップS65の処理で生成されたJava(R)オブジェクトに設定する。

【0322】

ステップS67において、変換ソフトウェア502は、ステップS65の処理で生成し、ステップS66の処理で各属性に値を設定した各Java(R)オブジェクトのそれぞれをインスタンスファイル（上述した図15のインスタンスファイル

301に相当するもの)にシリアルライズする。

【0323】

ステップS68において、変換ソフトウェア502は、XMLデータ511の中から対象シーンのECMASクリプトのソースコード512を抽出し、スクリプトファイル（上述した図15のスクリプトファイル303に相当するもの）として出力する。

【0324】

ステップS69において、変換ソフトウェア502は、ステップS67の処理でシリアルライズしたインスタンスファイル、ステップS68の処理で出力したスクリプトファイル、対象シーンに含まれる各コンポーネントのそれぞれに対応するコンポーネントのクラスファイル群514、および、対象シーンに組み込まれるテキスト/静止画等のファイル群513を、対象シーンに対応するシーンファイル292としてアーカイブする。

【0325】

即ち、変換ソフトウェア502は、図15に示されるような、対象シーンに対応するシーンファイル292を生成する。

【0326】

ステップS70において、変換ソフトウェア502は、全てのシーンが対象シーンとして選定されたか否かを判定する。

【0327】

ステップS70において、全てのシーンが対象シーンとして選定されていないと判定した場合、変換ソフトウェア502は、その処理をステップS62に戻し、まだ選択されていないシーンの中から、次の対象シーンを選択し、それ以降の処理を繰り返す。

【0328】

即ち、変換ソフトウェア502は、全てのシーンのそれぞれに対して、上述したステップS63乃至ステップS69の処理を施して、全てのシーンのそれぞれに対応するシーンファイル292を生成する。

【0329】

そして、変換ソフトウェア502は、最後のシーンに対応するシーンファイル292を生成すると、ステップS70において、全てのシーンが対象シーンとして選択されたと判定し、全てのシーンのそれぞれに対応するシーンファイル292からなるマルチメディアコンテンツ516と、ブラウザのアーカイブファイル291とを組み合わせ、配信ソフトウェア515を生成し、サーバ124に出力する。

#### 【0330】

その後、サーバ124が、その配信ソフトウェア515をネットワーク112（図9）を介して受信端末113（図9）に配信した場合、上述したように、受信端末113が、配信されたブラウザのアーカイブファイル291に対応するブラウザを利用して、配信されたマルチメディアコンテンツ516を再生させると、受信端末113のディスプレイ（図12の出力部187）には、図29に示されるような画像521が表示される。

#### 【0331】

このように、コンテンツ製作者517は、上述したオーサリングシステム441の代わりのソフトウェアとして、上述したテキストエディタ501と変換ソフトウェア502を無償、または、より安価な値段で入手することができ、かつ、そのようにして入手したテキストエディタ501と変換ソフトウェア502を容易に利用することができるので、オーサリングシステム441を利用しなくても、オーサリングシステム441により生成されるものと同一のフォーマット（図15に示されるフォーマット）で、マルチメディアコンテンツ516を含む配信ソフトウェア515を製作することが可能になる。

#### 【0332】

ただし、変換ソフトウェア502は、オーサリングシステム441と比較すると、マルチメディアコンテンツの生産性が低い。換言すると、オーサリングシステム441は、上述したように、以下の第1乃至第4の機能といった特有の機能を有するが、変換ソフトウェア502は、これらの機能を有していない。

#### 【0333】

即ち、第1の機能は、WYSIWYG (What You See Is What You Get) 環境を提供

する機能である。

【0334】

なお、ここでは、コンテンツの製作時に、最終的に得られるプレゼンテーションを視覚的に確認しながら編集を行うことが可能なユーザインターフェイス環境を、WYSIWYG環境と称している。WYSIWYG環境は、コンテンツに対して何らかの変更を施すと、その効果を視覚的に即座に確認することができるので、コンテンツ製作の効率が大幅に向上するという効果を奏することが可能である。

【0335】

第2の機能は、コンテンツの構成、編集対象コンポーネントのカスタマイズ情報等、コンテンツの製作に必要な各種情報のリストやツリーといった分かりやすい形での表示を提供する機能である。

【0336】

第3の機能は、コンポーネントのカスタマイズ支援の機能である。即ち、第3の機能は、コンポーネントの属性値を入力するとき、テキストフィールド、チェックボックス、スライダー等のGUIコンポーネントを利用することにより、入力サポートを行うとともに、ユーザの入力値が正しい範囲にあるか否か等のチェックを行う機能である。

【0337】

第4の機能は、コピー、ペースト、操作の取り消し、やり直し等の処理を提供する機能である。

【0338】

また、テキストエディタ501は、ユーザが、定められたテキストフォーマットに準じて正確にコンテンツを記述する必要があり、記述ミスがあると変換時にエラーになるという特徴を有している。従って、マルチメディアコンテンツの生産性が高いソフトウェアとは言い難い。

【0339】

このように、テキストエディタ501と変換ソフトウェア502を利用する製作環境と、オーサリングシステム441を利用するコンテンツ製作環境は相互に補完し合う関係にあると考えることができる。

## 【0340】

なお、上述したサーバ124が配信する配信ソフトウェア（例えば、図13の配信ソフトウェア242、および、図27の配信ソフトウェア515）自体のフォーマットに、XML等を利用したテキスト形式を採用することも可能であるが、この場合、データサイズが大きくなり、ロードに時間がかかるといった新たな課題が発生してしまう。従って、サーバ124が配信する配信ソフトウェアは、上述したように、バイナリ形式のファイルとして配信されたほうが好適である。

## 【0341】

以上、本実施の形態においては、上述したように、コンポーネントベースのアーキテクチャを採用しているので、マルチメディアコンテンツにコンテンツ独自の機能を従来よりも容易に組み込むことができる（拡張性が高いといえる）。

## 【0342】

また、本実施の形態においては、3Dをベースとしたプレゼンテーションを実現するので、分かり易く魅力的なマルチメディア視聴の環境を実現することができる。

## 【0343】

さらに、本実施の形態においては、オーサリングツール上で、任意のコンポーネントをコンテンツに組込み、グラフィカルな環境で各々のコンポーネントをカスタマイズし、かつ、スクリプト言語により容易にコンポーネントを制御できる環境を提供するので、デザイナーやコンテンツクリエイターが従来よりも容易に高機能なコンテンツを製作することができる（生産性が高いといえる）。

## 【0344】

さらに、本実施の形態においては、テキストデータとして記述されたコンテンツのフォーマットを、上述したオーサリングシステムにより製作されたものと同じのフォーマットに変換する変換ソフトウェアが用意されているので、コンテンツ製作者は、一般的なテキストエディタを利用して、コンテンツの製作や編集が可能になる。このようなテキストエディタは、OSに付属している場合や、フリーウェアやシェアウェアである場合が多く、無償または安価に入手可能である。

## 【0345】

また、この変換ソフトウェアは、上述したオーサリングシステムと比較して、極めて安価に開発することが可能であるので、コンテンツ製作者（ユーザ）のコストの負担を少なくすることができる。さらに、テキストでコンテンツを記述するための文法として独自のものを利用することも可能であるが、XML等のオープンな規格に準じた方式を採用することにより、変換ソフトウェアの開発コストをさらに抑制し、コンテンツの製作者（ユーザ）にとってさらに利用しやすいツールとすることが可能になる。

## 【0346】

さらにまた、本実施の形態においては、提示の一意性を保証し、かつ、矢印または決定キーによるユーザーインターフェースを採用するので、パーソナルコンピュータを含む様々なデジタル家電製品において、従来よりも簡便でかつ共通した操作性を提供することができるとともに、パーソナルコンピュータを含む様々なデジタル家電製品に対して同一のコンテンツを配信できる。

## 【0347】

このように、本実施の形態は、上述した第1乃至第6の課題を解決することができる。

## 【0348】

なお、上述した本実施の形態の各種のソフトウェアを構成するプログラムは、ネットワークや記録媒体からインストールされる。

## 【0349】

この記録媒体は、図10、図11、または図12に示されるように、装置本体とは別に、ユーザにプログラムを提供するために配布される、プログラムが記憶されているリムーバブル記録媒体（磁気ディスク（フロッピーディスクを含む）、光ディスク（CD-ROM(Compact Disk-Read Only Memory), DVD(Digital Versatile Disk)を含む）、光磁気ディスク（MD (Mini-Disk) を含む）、もしくは半導体メモリ）151、171、または191などよりなるパッケージメディアにより構成されるだけでなく、装置本体に予め組み込まれた状態でユーザに提供される、プログラムが記憶されているROM142、162、または182や、記憶部148、168、または188に含まれるハードディスクなどで構成される。

【0350】

なお、本明細書において、記録媒体に記録されるプログラムを記述するステップは、記載された順序に沿って時系列的に行われる処理はもちろん、必ずしも時系列的に処理されなくとも、並列的あるいは個別に実行される処理をも含むものである。

【0351】

さらに、本明細書において、システム（オーサリングシステム441を除く）とは、複数の装置や処理部により構成される装置全体を表すものである。

【0352】

【発明の効果】

以上のごとく、本発明によれば、マルチメディアコンテンツを取り扱うことができる。さらに、本発明によれば、高機能なマルチメディアコンテンツを容易に取り扱うことができる。

【図面の簡単な説明】

【図1】

W.W.Wの動作原理を説明するW.W.Wシステムの構成例を示す斜視図である。

【図2】

HTML文書の例とその表示例を示す図である。

【図3】

提示の一意性が保証されない場合の、図2のHTML文章の表示例を示す図である。

【図4】

提示の一意性が保証されない場合の、図2のHTML文章の表示の他の例を示す図である。

【図5】

提示の一意性が保証される場合の、図2のHTML文章の表示例を示す図である。

【図6】

HTML文書の配信方式を説明する図である。

【図7】



Java(R)アプリケーションの配信方式を説明する図である。

【図 8】

上下左右及び決定キーの構成例を説明する図である。

【図 9】

本発明が適用される情報配信システムの構成例を示す図である。

【図 1 0】

図 9 の情報配信システムのマルチメディアコンテンツ製作装置の構成例を示すブロック図である。

【図 1 1】

図 9 の情報配信システムのサーバの構成例を示すブロック図である。

【図 1 2】

図 9 の情報配信システムの受信端末の構成例を示すブロック図である。

【図 1 3】

マルチメディアコンテンツを再生する仕組みを説明する図である。

【図 1 4】

受信端末側のソフトウェアスタックを示す図である。

【図 1 5】

図 1 3 の配信ソフトウェアの構成例を示す図である。

【図 1 6】

図 1 3 のブラウザとコンポーネントのインタフェースである MComponent パッケージの構成例を示す図である。

【図 1 7】

図 1 3 のブラウザのフォーカス遷移の例を説明する図である。

【図 1 8】

図 1 3 のブラウザのメディア同期機構のシーケンスを説明する図である。

【図 1 9】

図 1 3 のブラウザの構成例を示すブロック図である。

【図 2 0】

図 1 3 のブラウザにおけるコンポーネントのライフサイクルを説明する状態遷

移図である。

【図21】

図13のブラウザのシーン起動処理を説明するフローチャートである。

【図22】

図13のブラウザのシーン終了処理を説明するフローチャートである。

【図23】

図13のブラウザのイベント配布機構のシーケンスを説明する図である。

【図24】

図14のアプリケーションマネージャの処理を説明するフローチャートである。

【図25】

オーサリングシステムを起動した場合の表示例を示す図である。

【図26】

図25のオーサリングシステムにより製作されたマルチメディアコンテンツの再生例を示す図である。

【図27】

変換ソフトウェアを実装する図9のマルチメディアコンテンツ制作装置のソフトウェアの構成例を示す図である。

【図28】

図27の変換ソフトウェアの処理を説明するフローチャートである。

【図29】

図27の変換ソフトウェアによりフォーマットが変換されたマルチメディアコンテンツが、図9の受信装置に配信され、再生された例を示す図である。

【図30】

図27のXMLデータ（XMLフォーマットのマルチメディアコンテンツ）の例を示す図である。

【図31】

図27のXMLデータ（XMLフォーマットのマルチメディアコンテンツ）の例を示す図である。

【図 3 2】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 3】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 4】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 5】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 6】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 7】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 8】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 3 9】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 0】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 1】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

す図である。

【図 4 2】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 3】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 4】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 5】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 6】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 7】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 8】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 4 9】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 0】

図 2 7 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 1】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 2】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 3】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 4】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 5】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 6】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 7】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 8】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 5 9】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

【図 6 0】

図 27 の XML データ (XML フォーマットのマルチメディアコンテンツ) の例を示す図である。

## 【図61】

図27のXMLデータ（XMLフォーマットのマルチメディアコンテンツ）の例を示す図である。

## 【図62】

図27のDTDの例を示す図である。

## 【図63】

図27のDTDの例を示す図である。

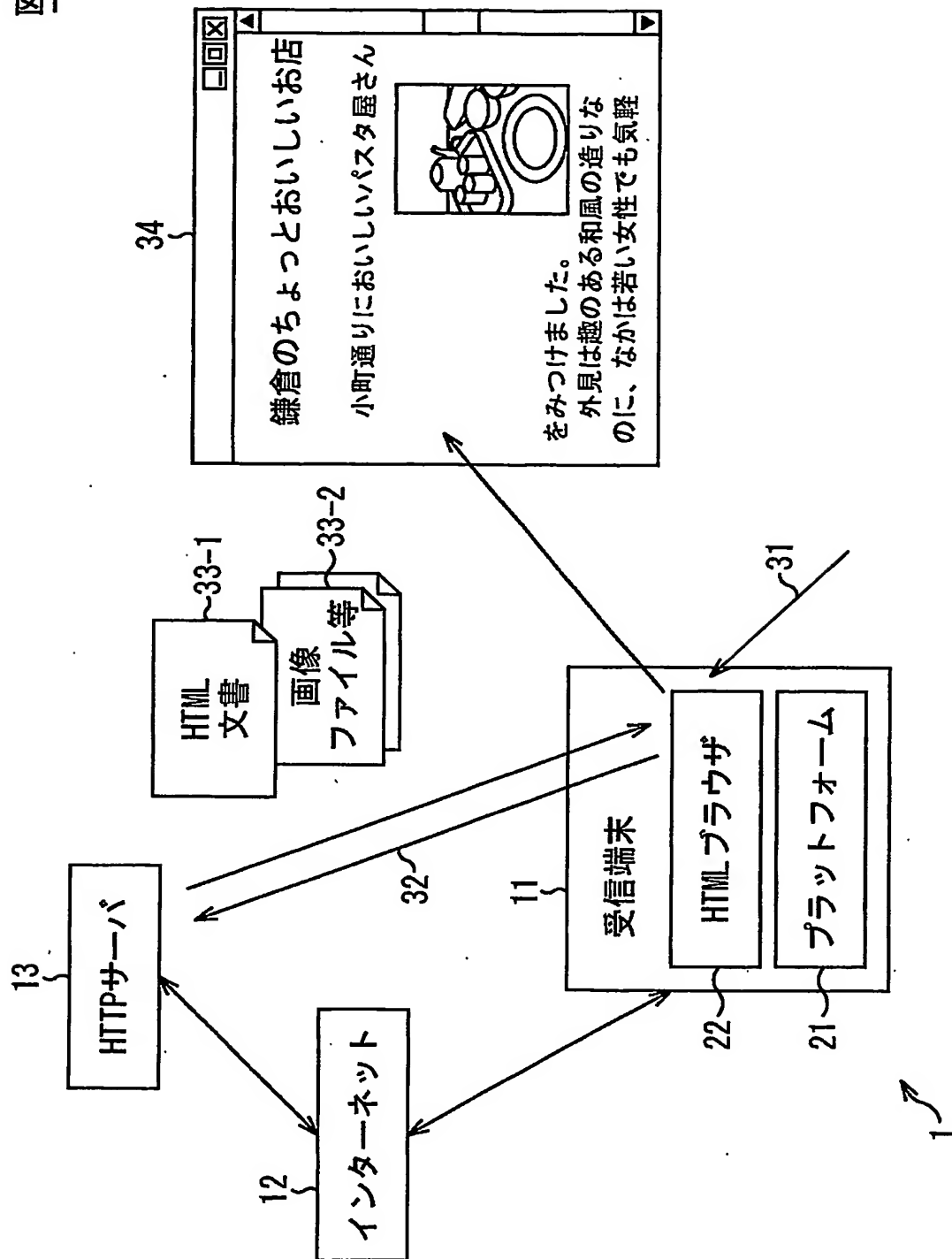
## 【符号の説明】

91 キー, 101 情報配信システム, 112 ネットワーク, 113 受信端末, 123 マルチメディアコンテンツ製作装置, 124 サーバ, 131 マルチメディアコンテンツおよびブラウザ131, 141, 161, 181 CPU, 243 配信ソフトウェア, 251 ブラウザ, 252 マルチメディアコンテンツ, 271 アプリケーションマネージャ, 292 シーンファイル, 361 制御部, 362 ロード部, 363 シーンデータ部, 364 描画部, 365 ECMAスクリプト処理部, 366 コンポーネントローダ, 368 コンポーネント群, 369 スクリプトデータ, 373 3D描画エンジン, 370 拡張関数, 371 スクリプトインタプリタ, 372 イベント/フォーカス管理部, 441 オーサリングシステム, 501 テキストエディタ, 502 変換ソフトウェア, 511 XMLデータ, 515 配信ソフトウェア, 516 マルチメディアコンテンツ

【書類名】 図面

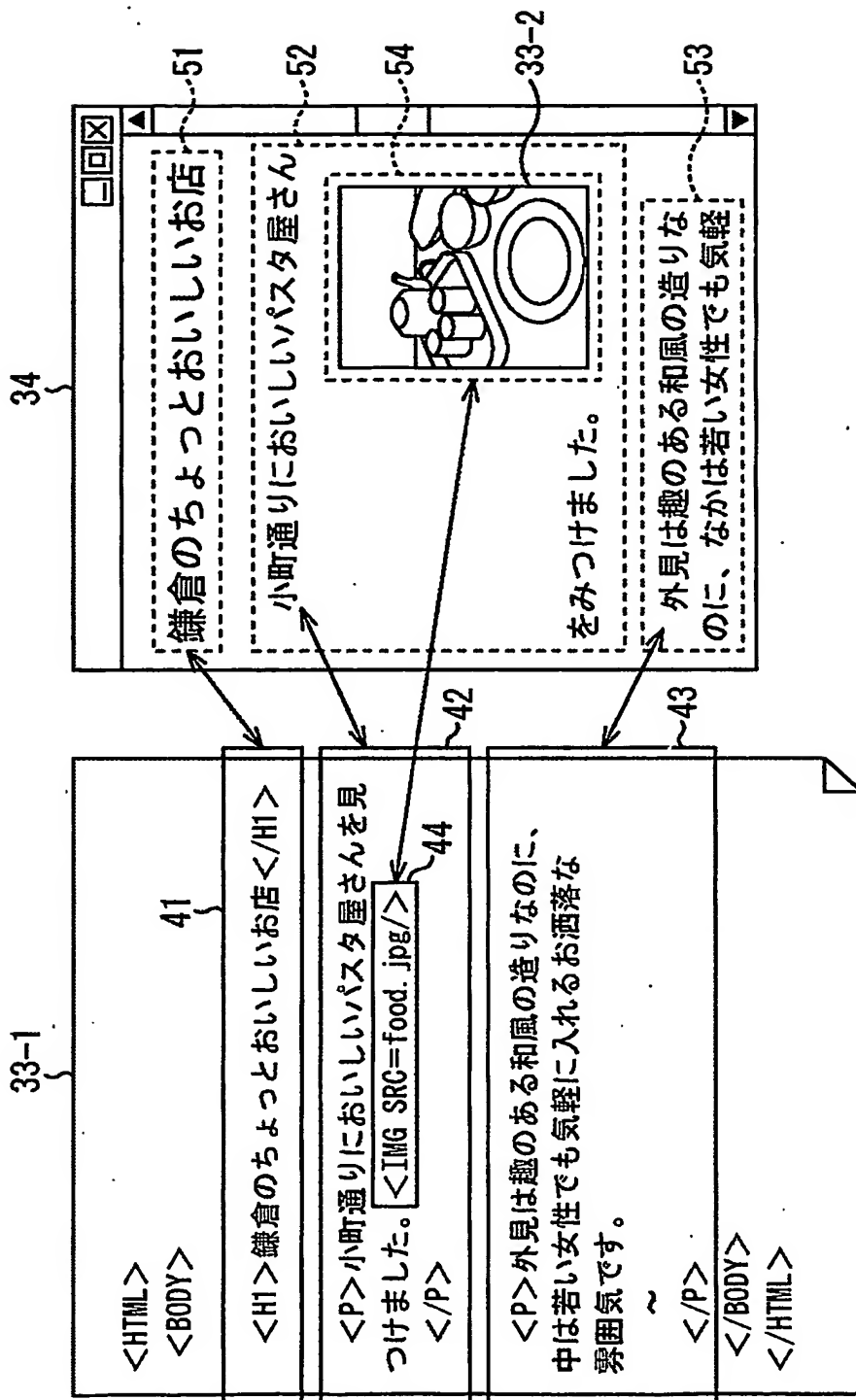
【図 1】

図1



【図2】

図2

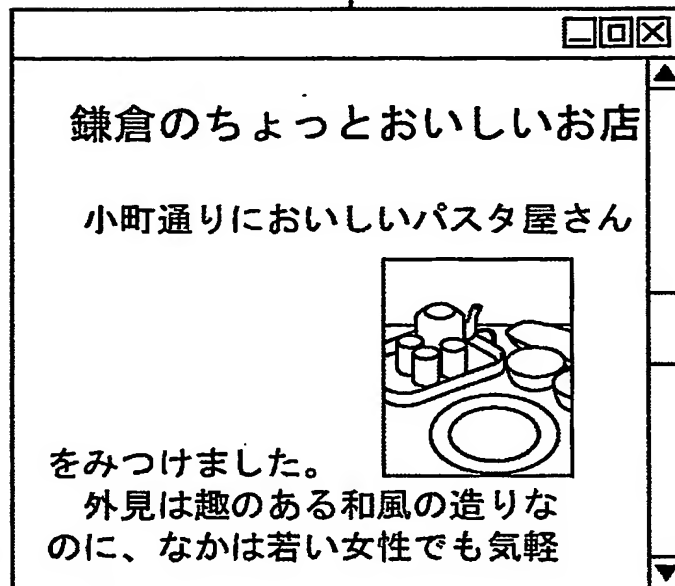




【図3】

図3

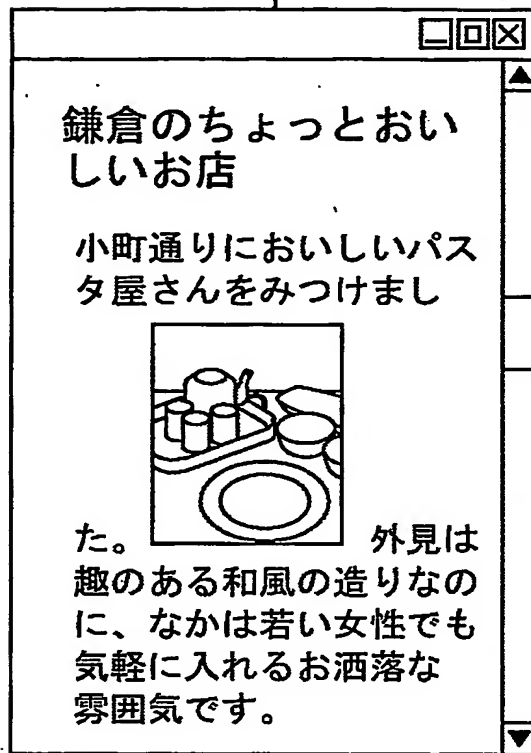
34-1



【図4】

図4

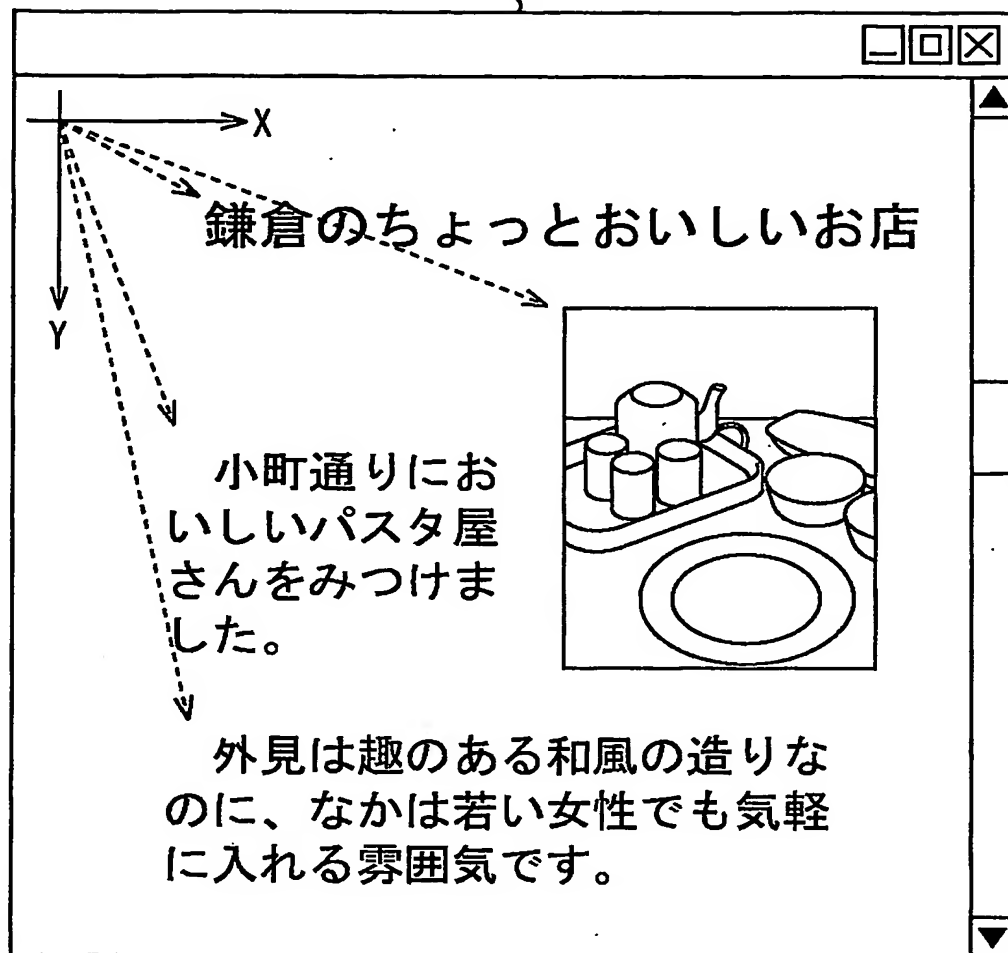
34-2



【図5】

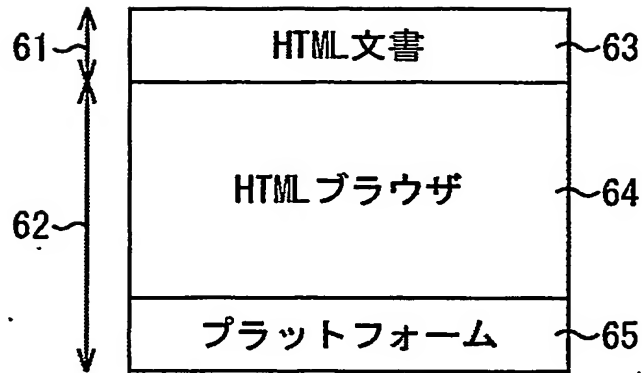
図5

34-3



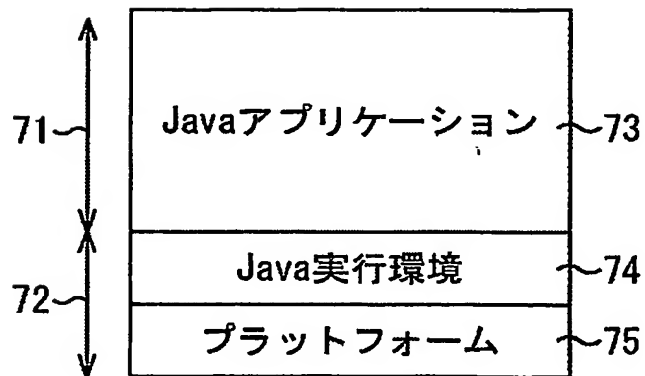
【図 6】

図6



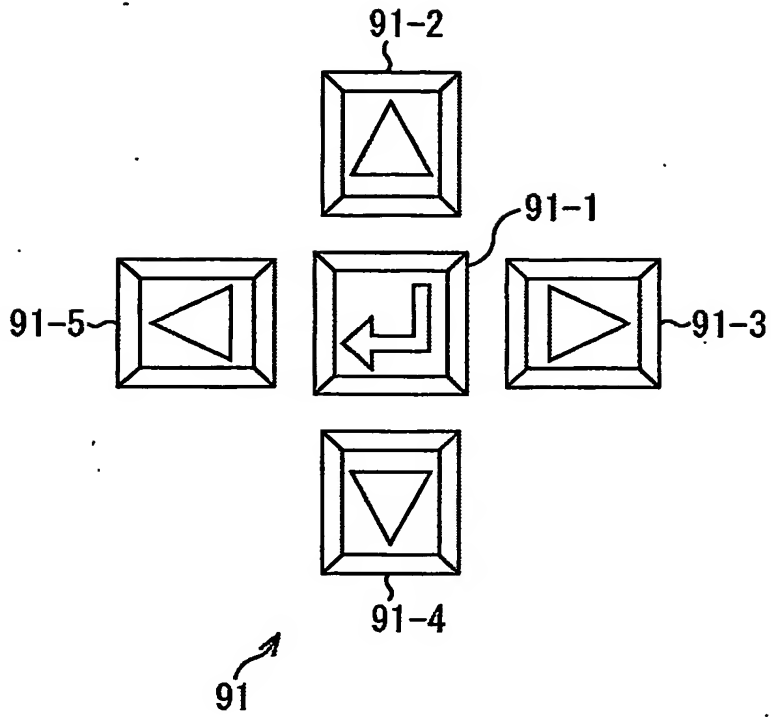
【図 7】

図7



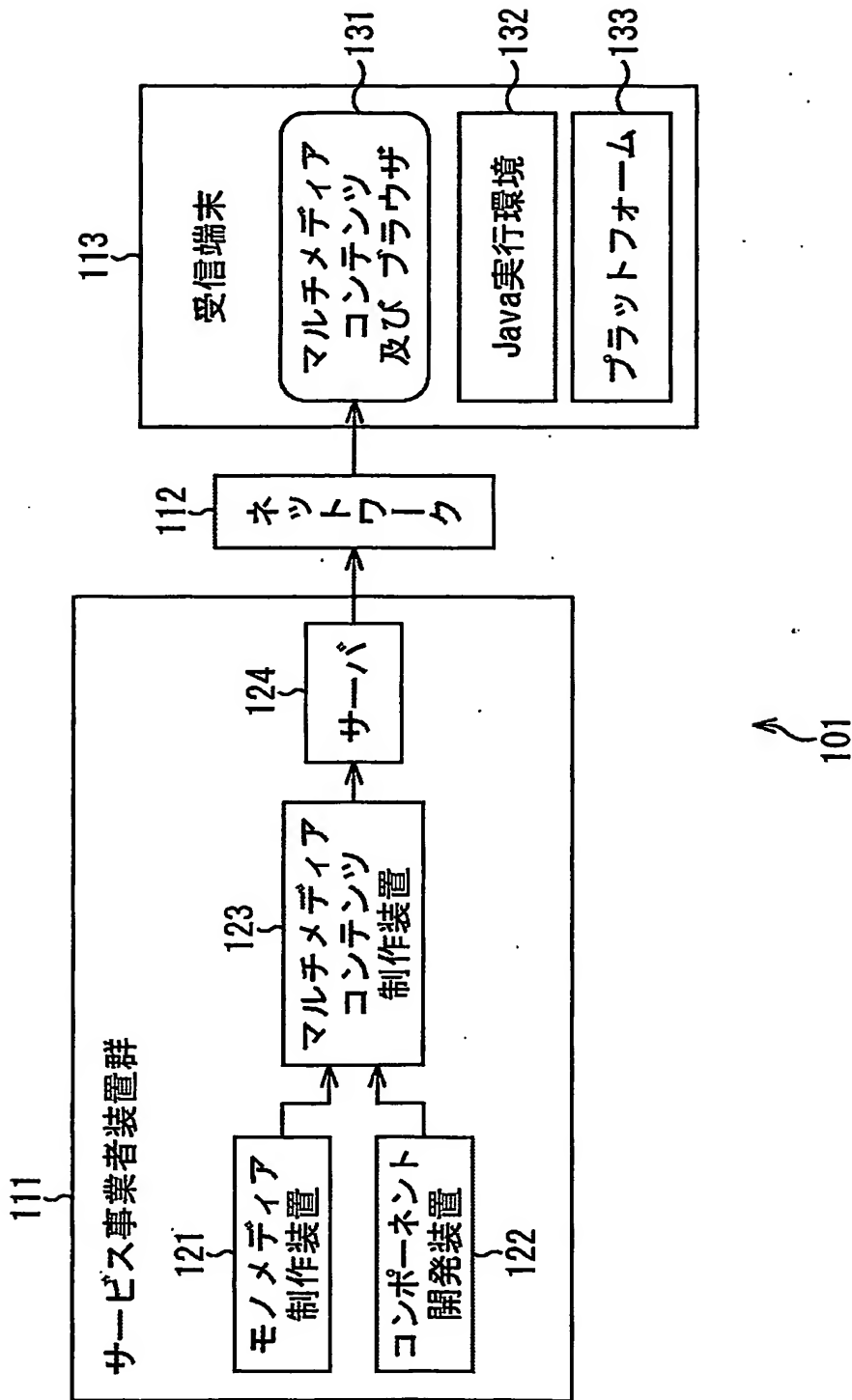
【図 8】

図 8



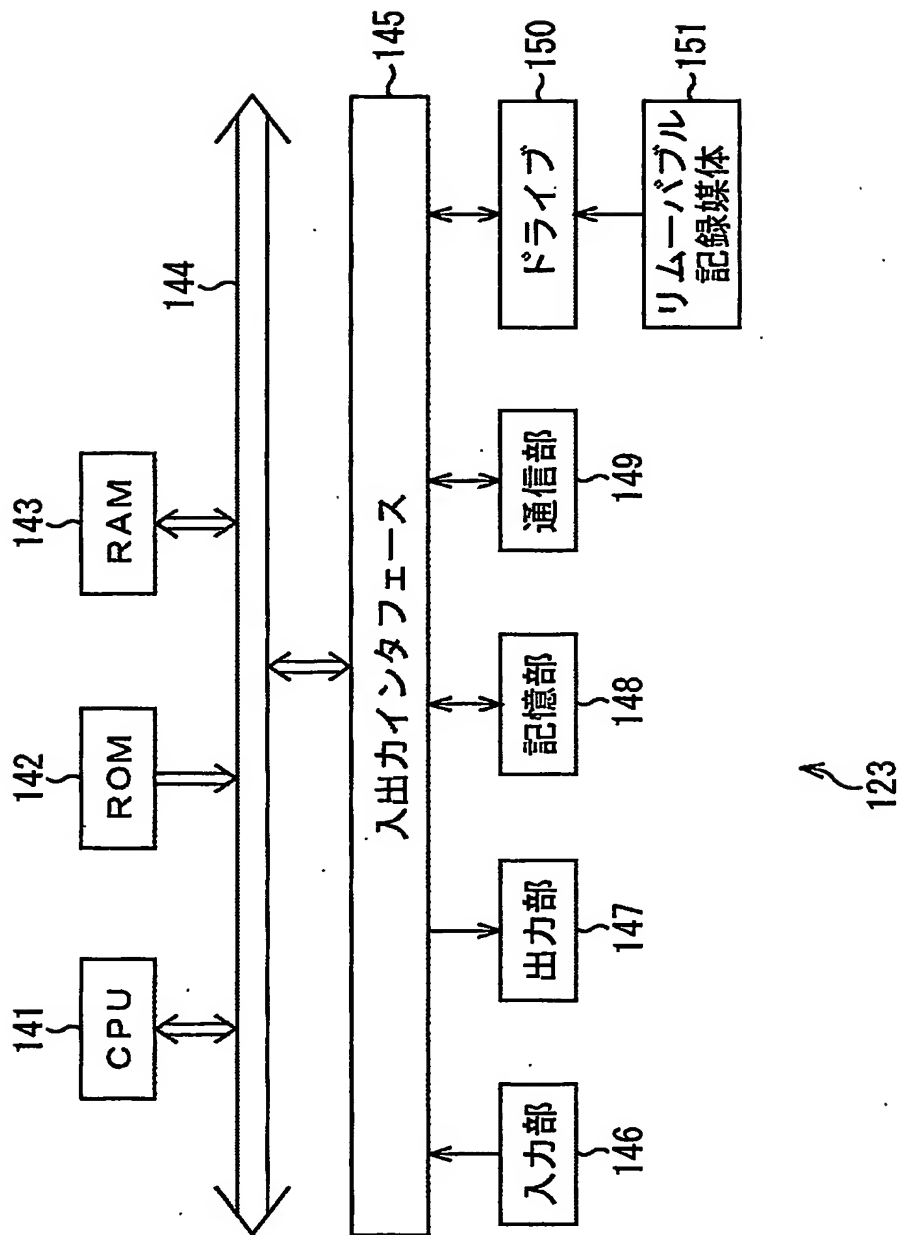
【図9】

図9



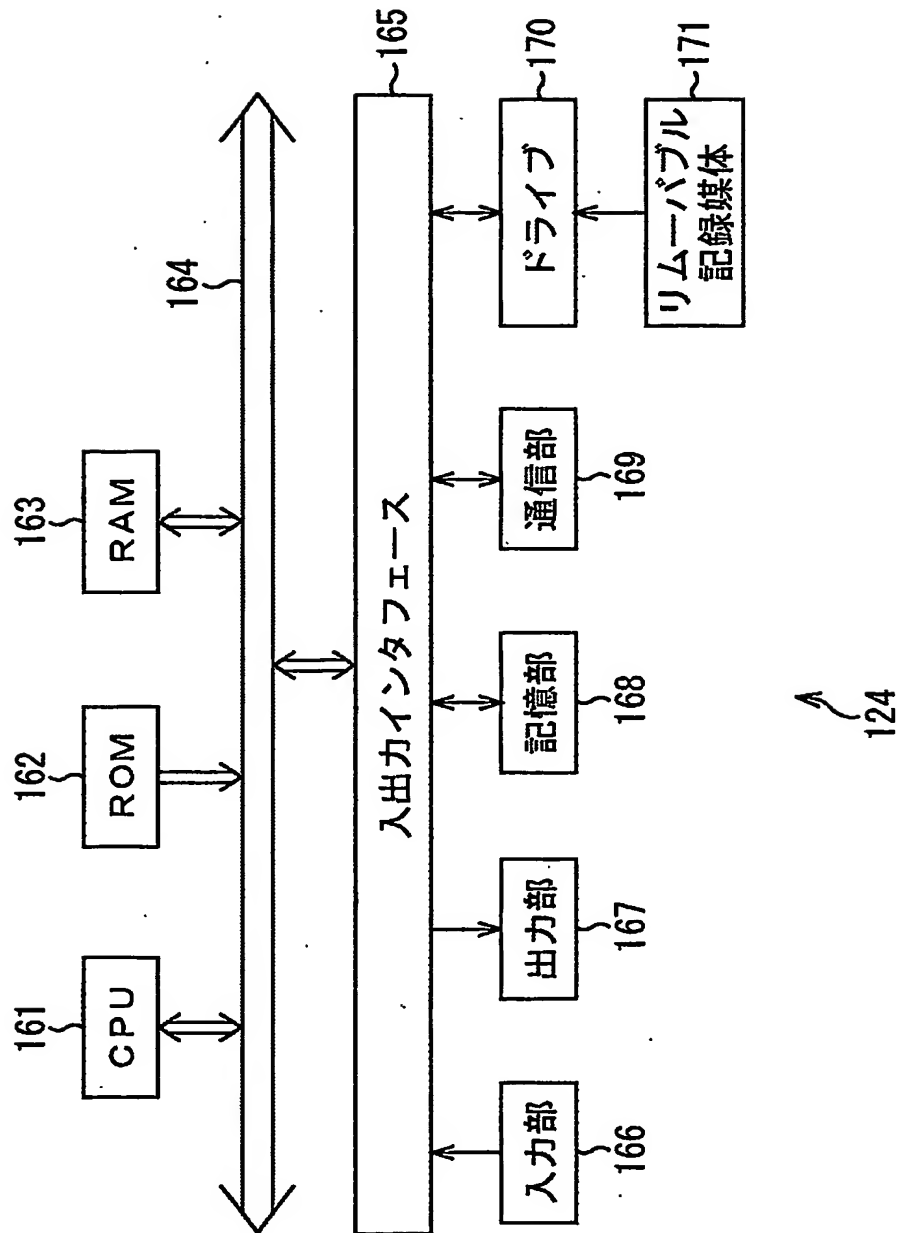
【図10】

図10



【図 11】

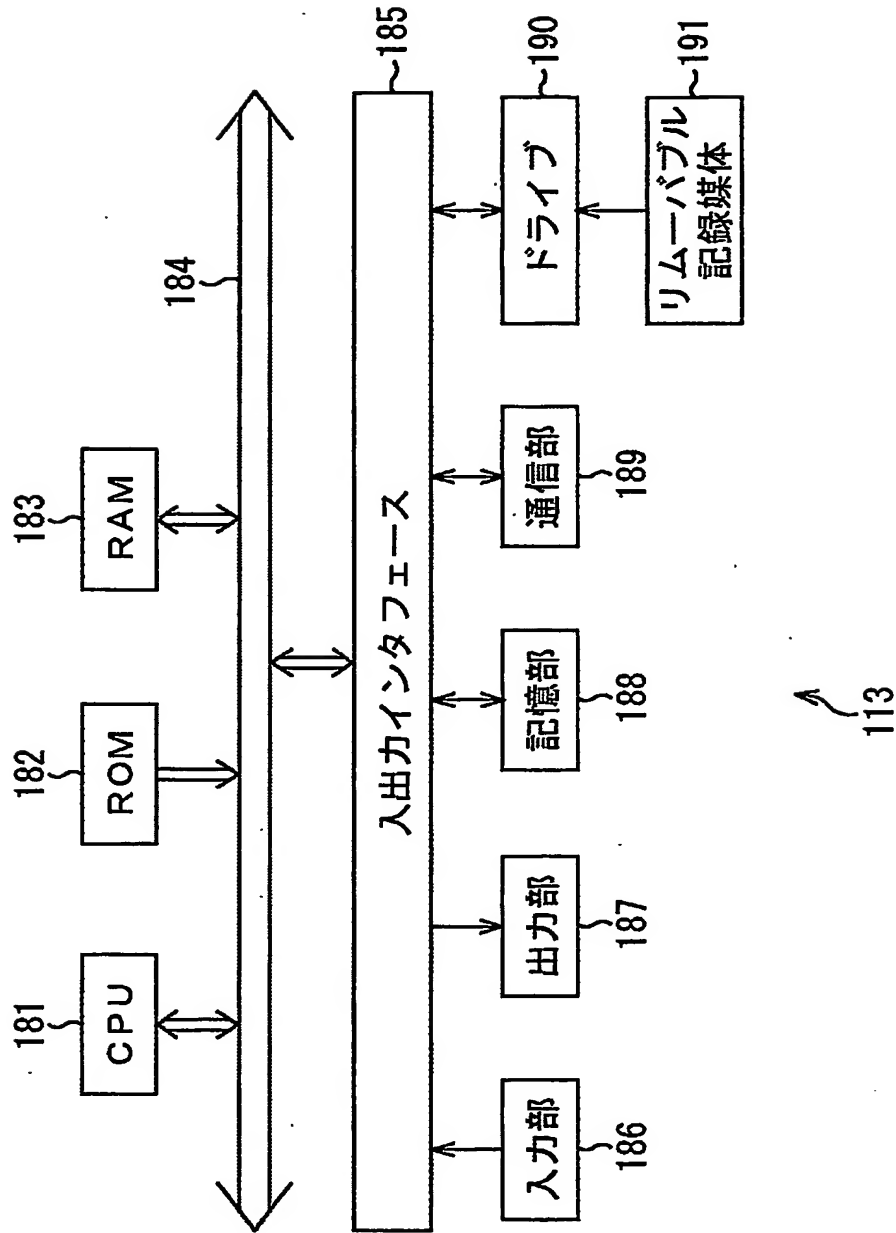
図11





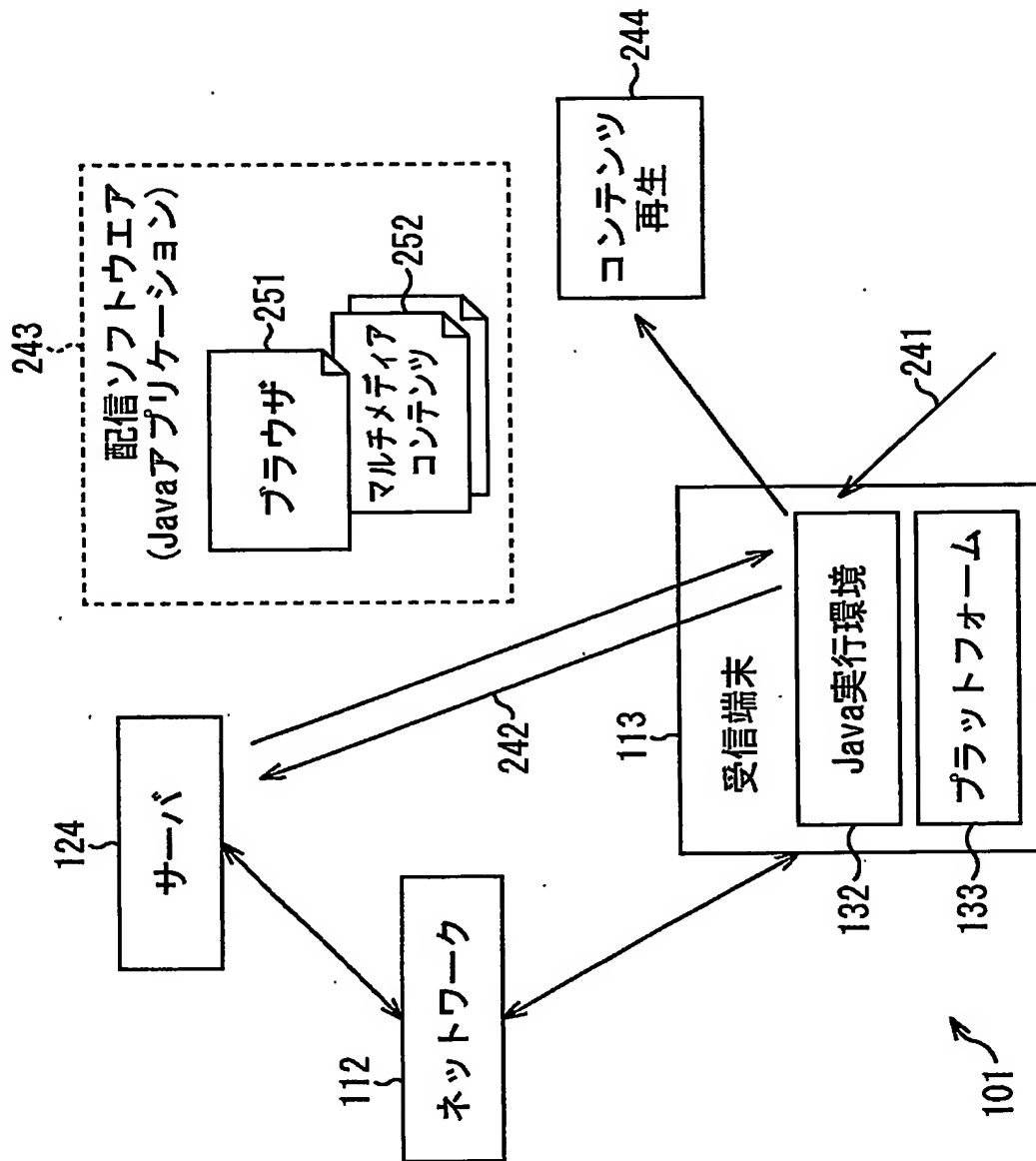
【図12】

図12



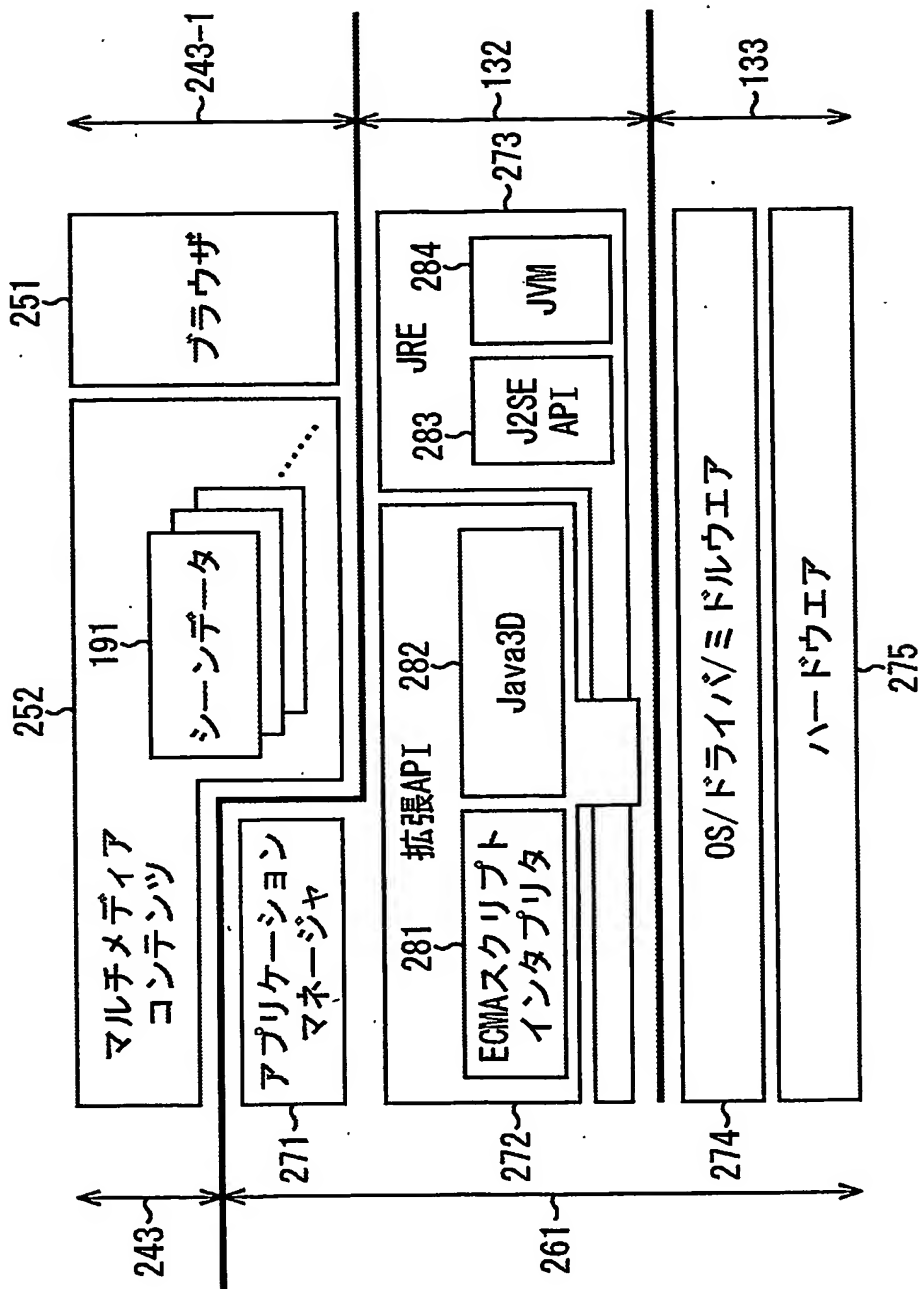
【図13】

図13



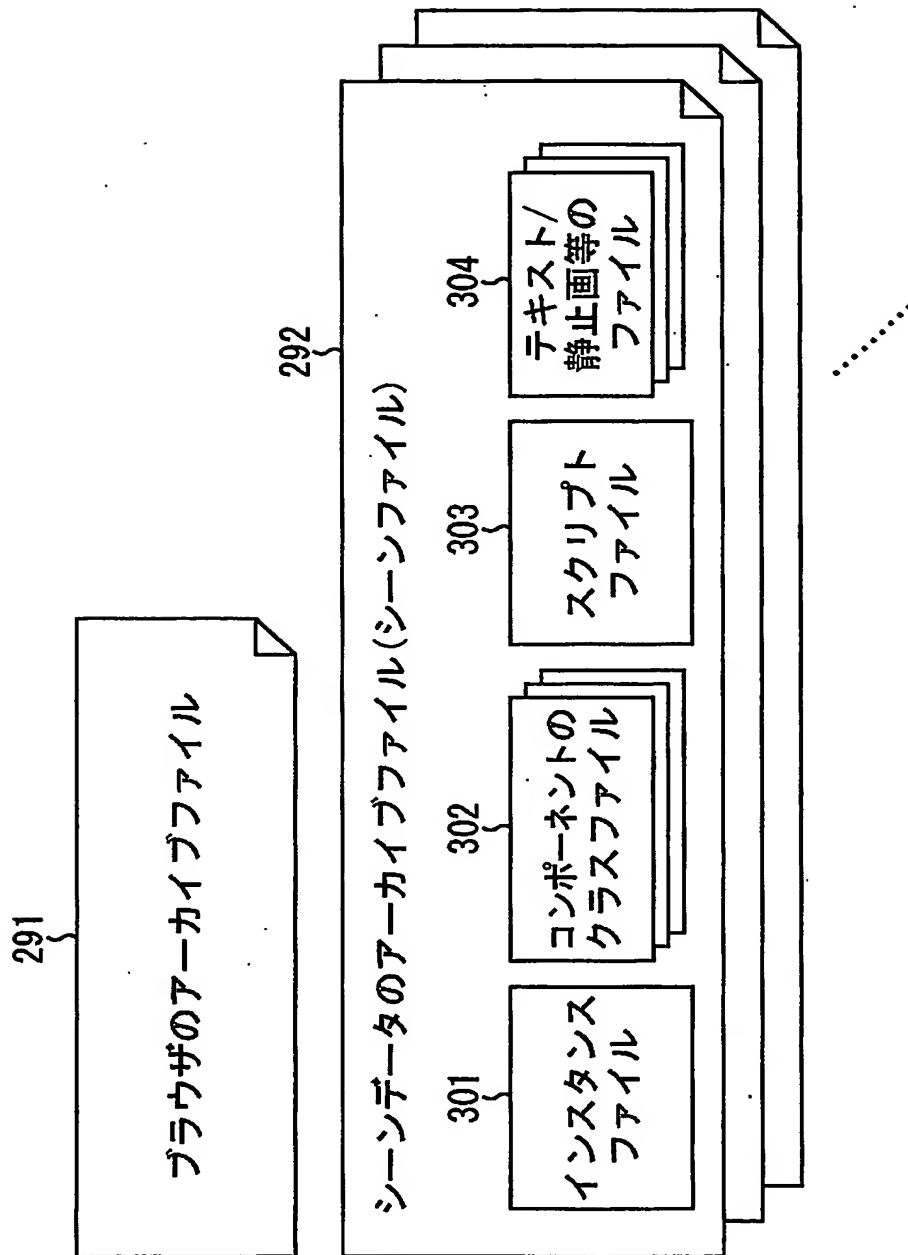
【図14】

図14



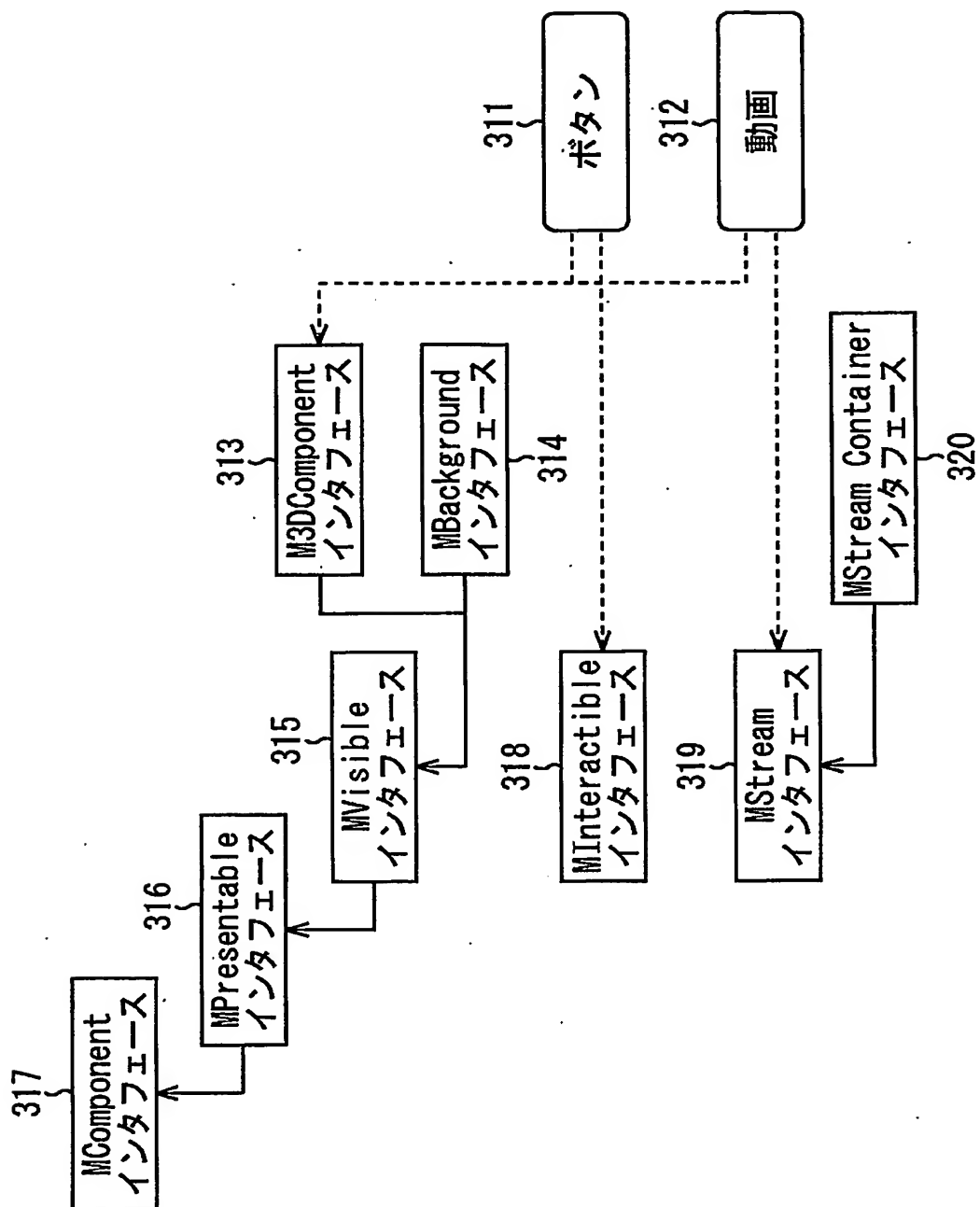
【図15】

図15



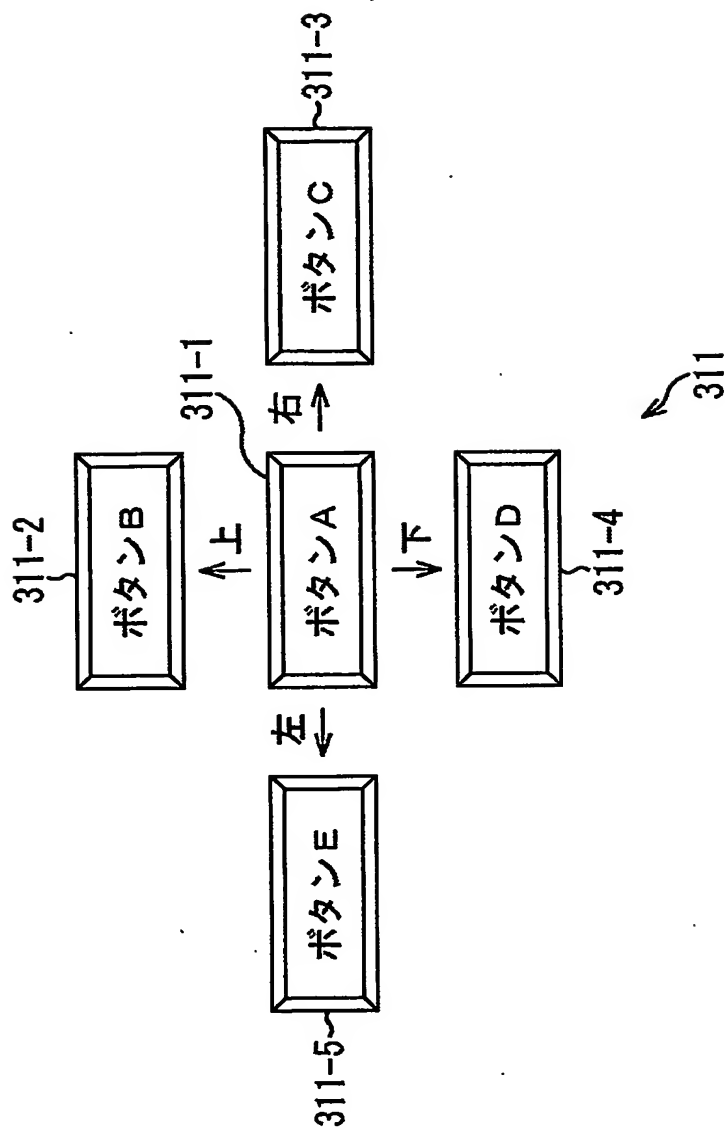
【図16】

図16



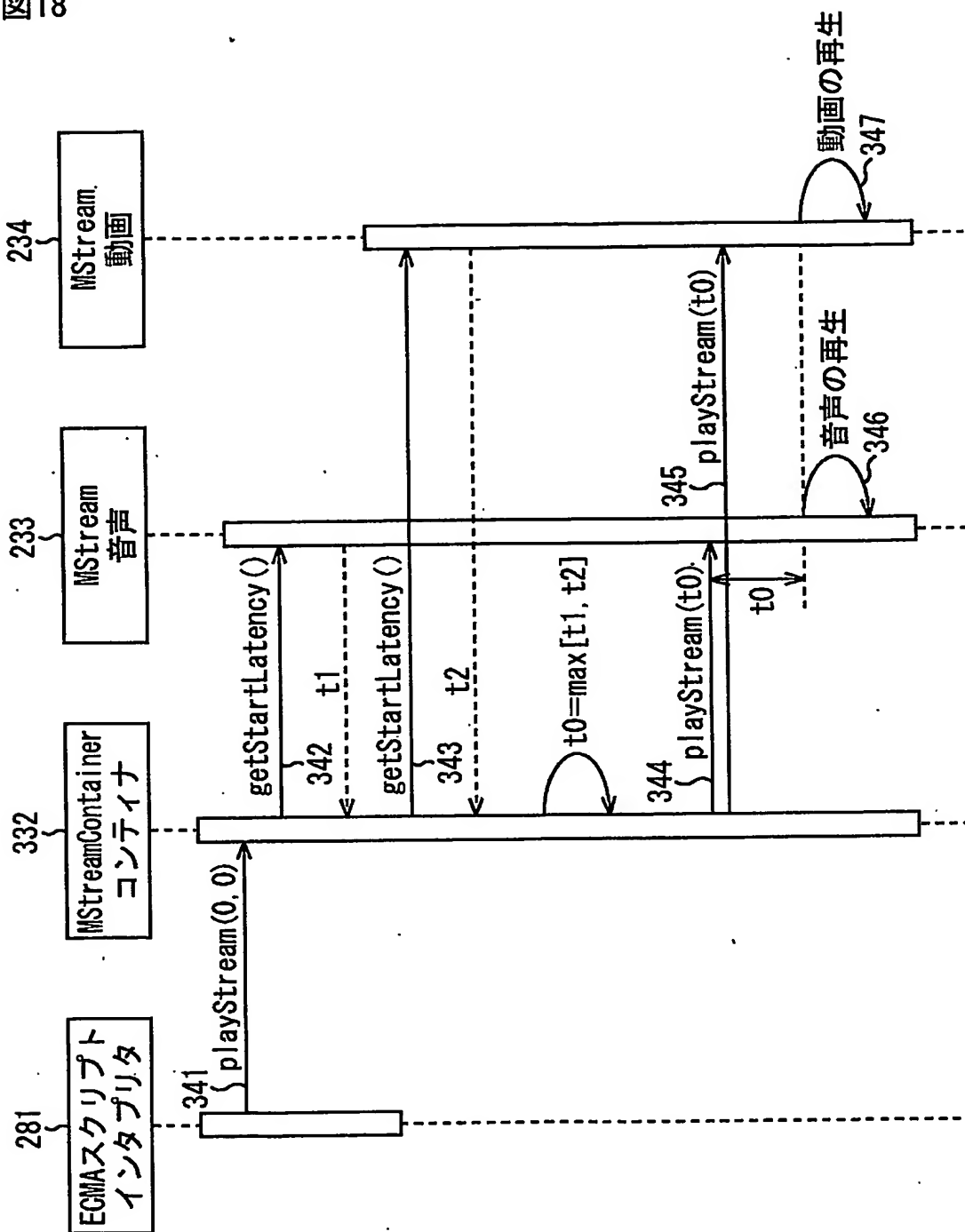
【図 17】

図 17



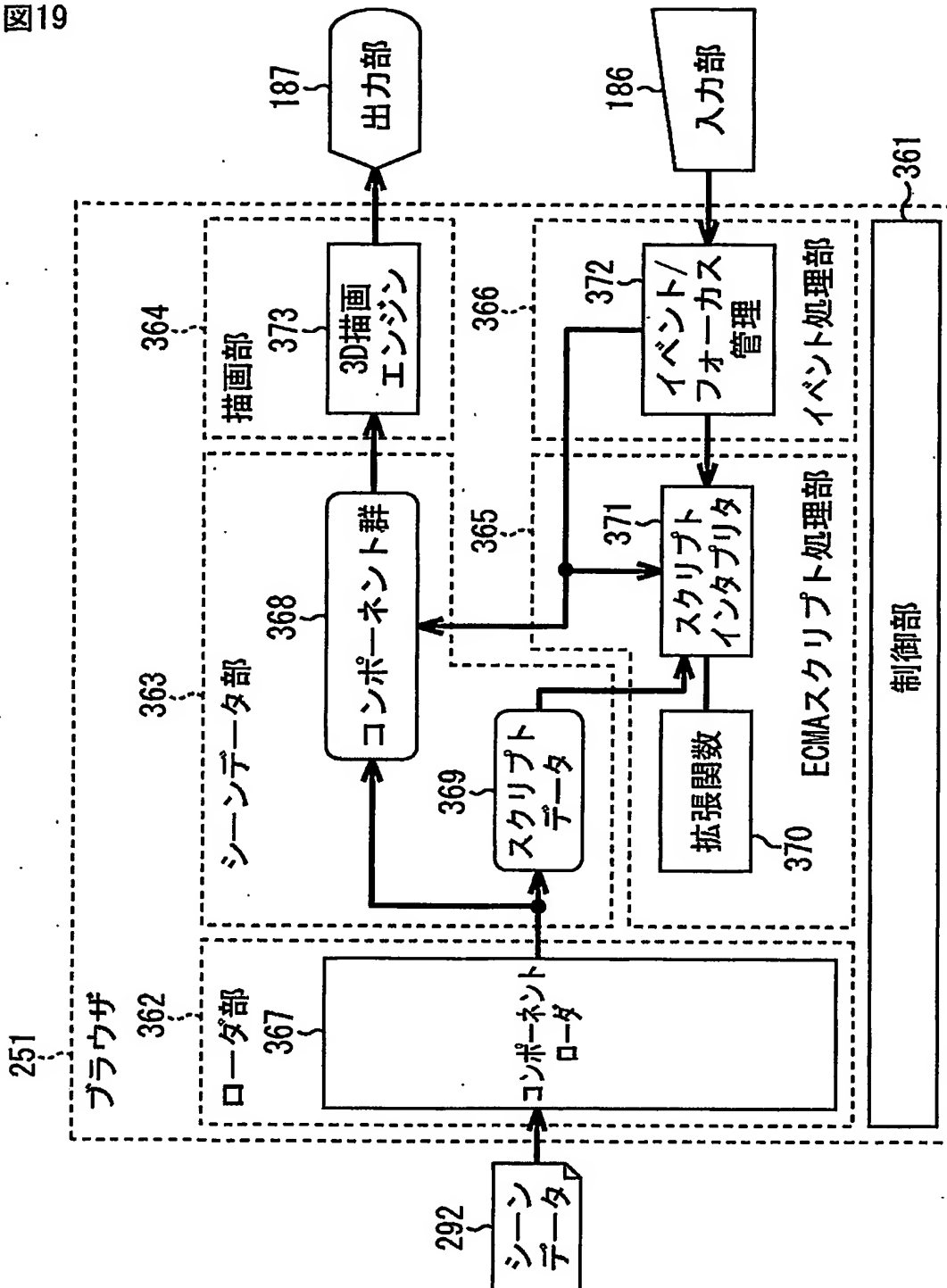
【図18】

図18



【図19】

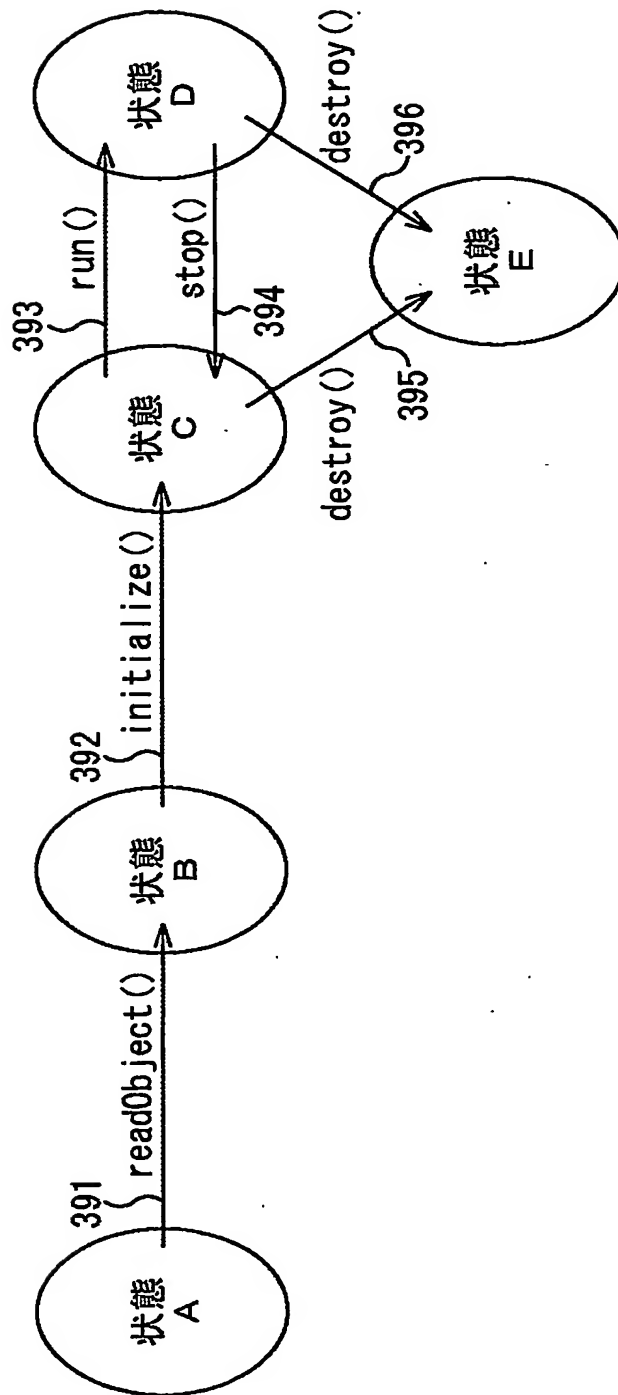
図19





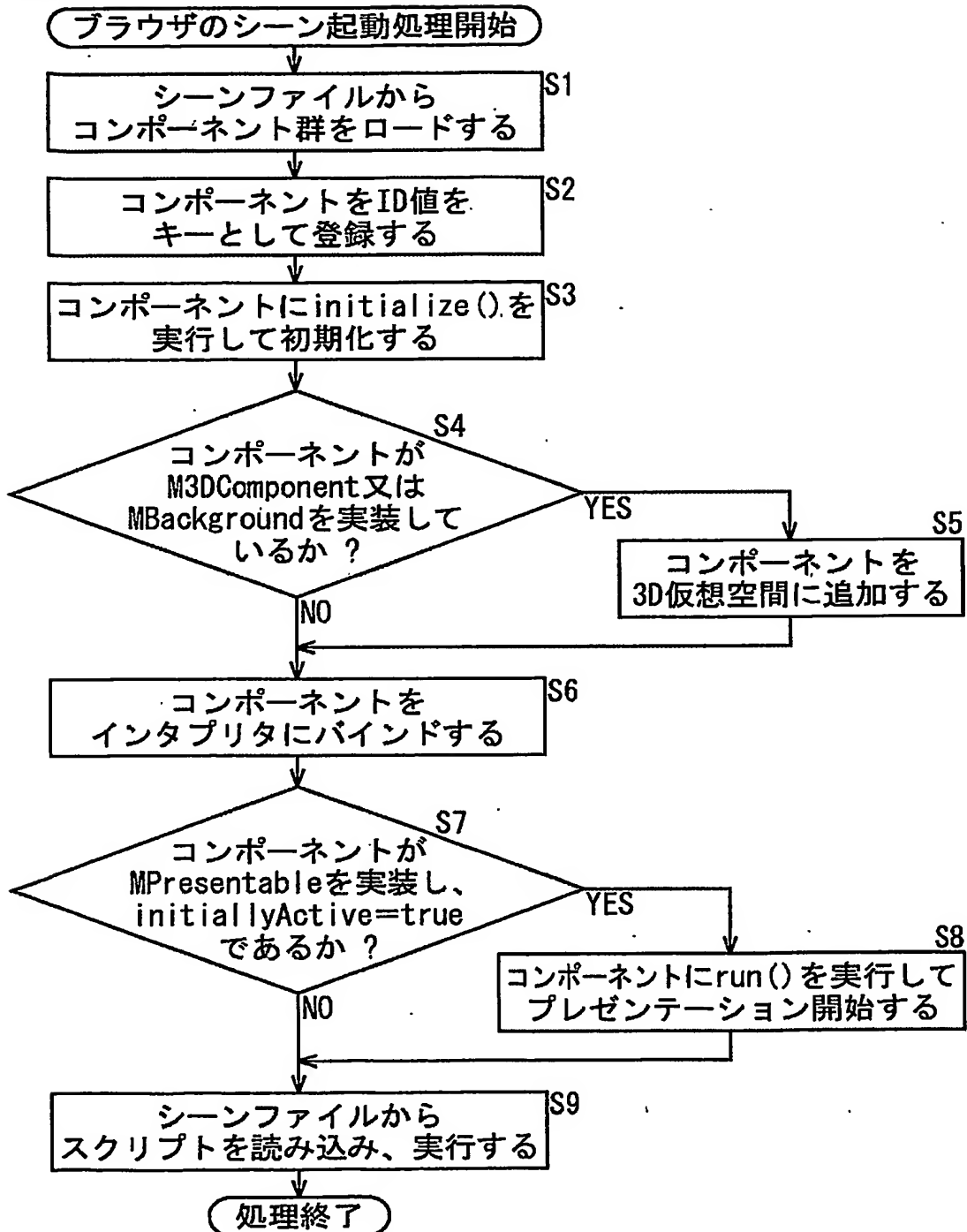
【図20】

図20



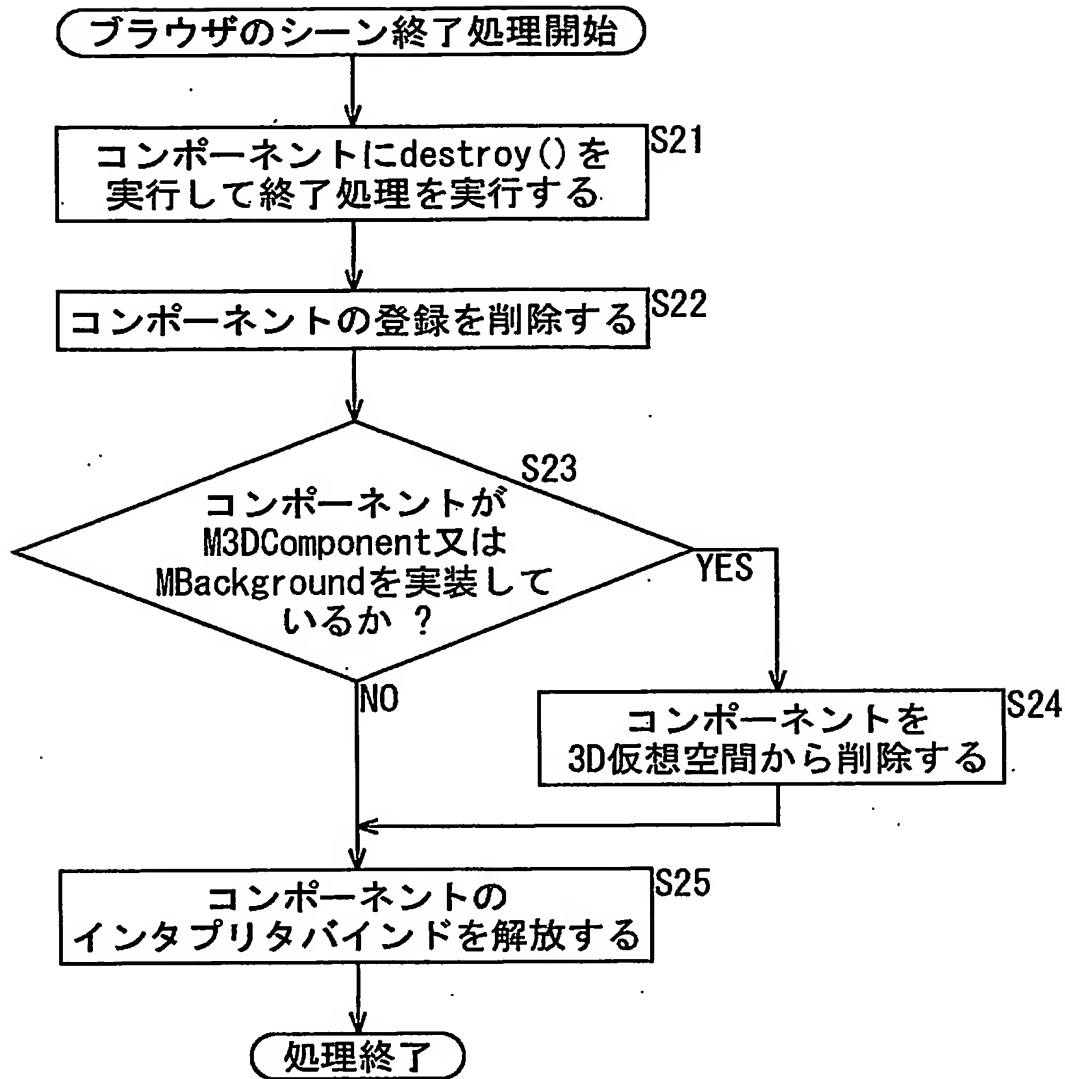
【図21】

図21



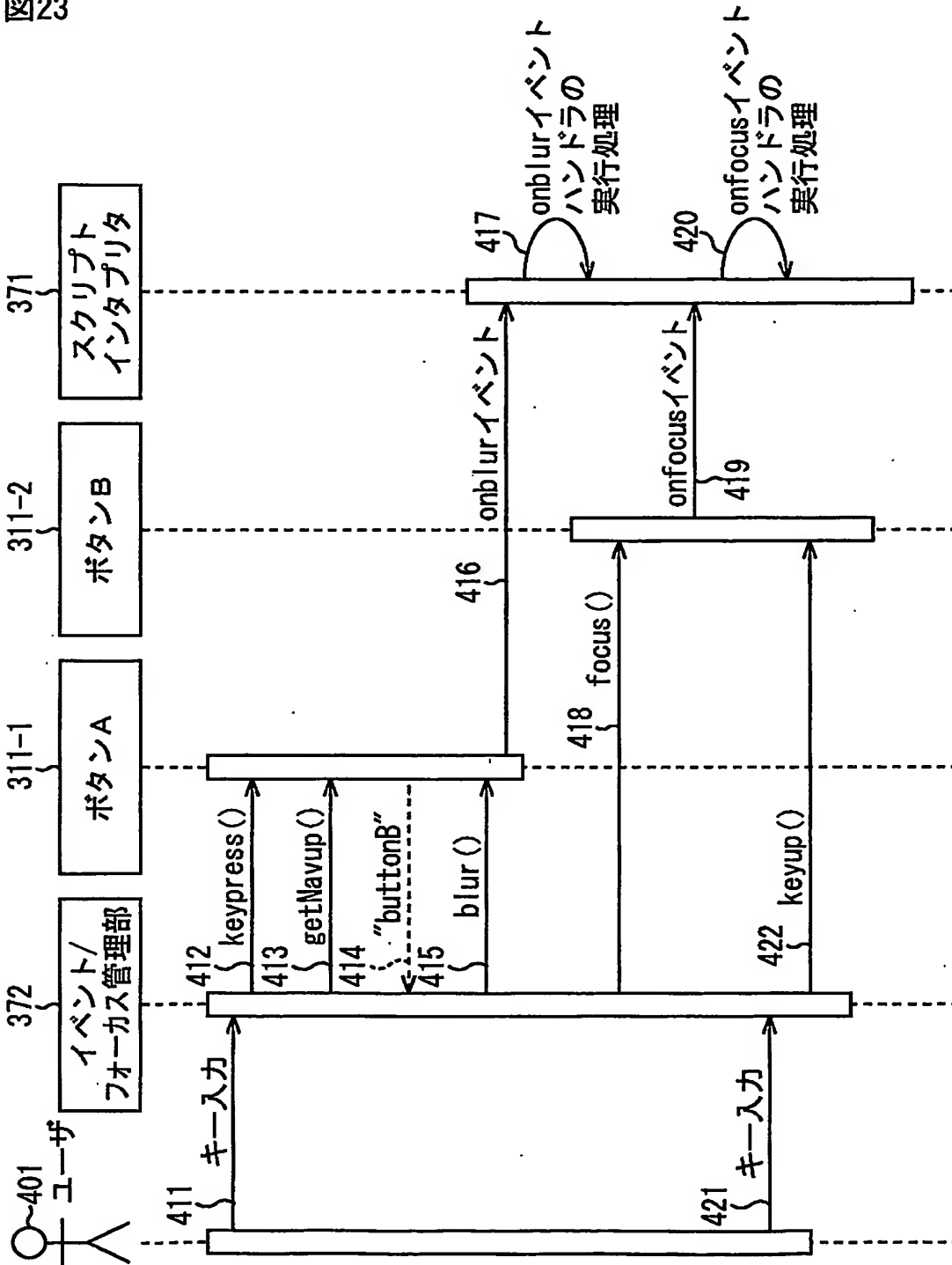
【図 2 2】

図22



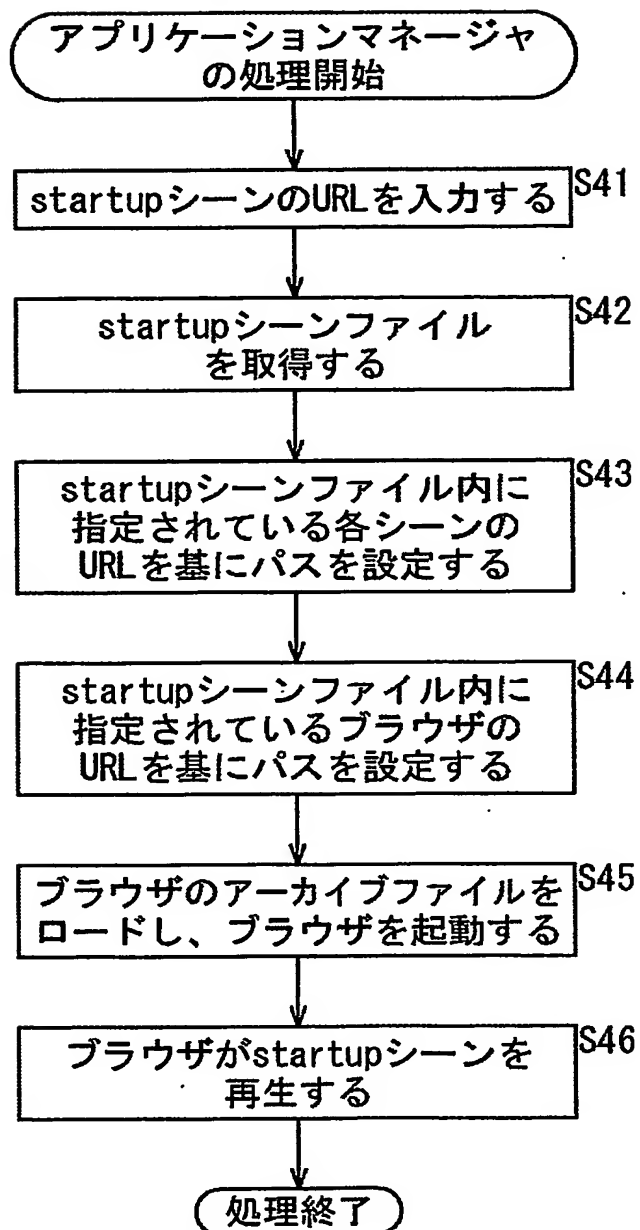
【図 23】

図23



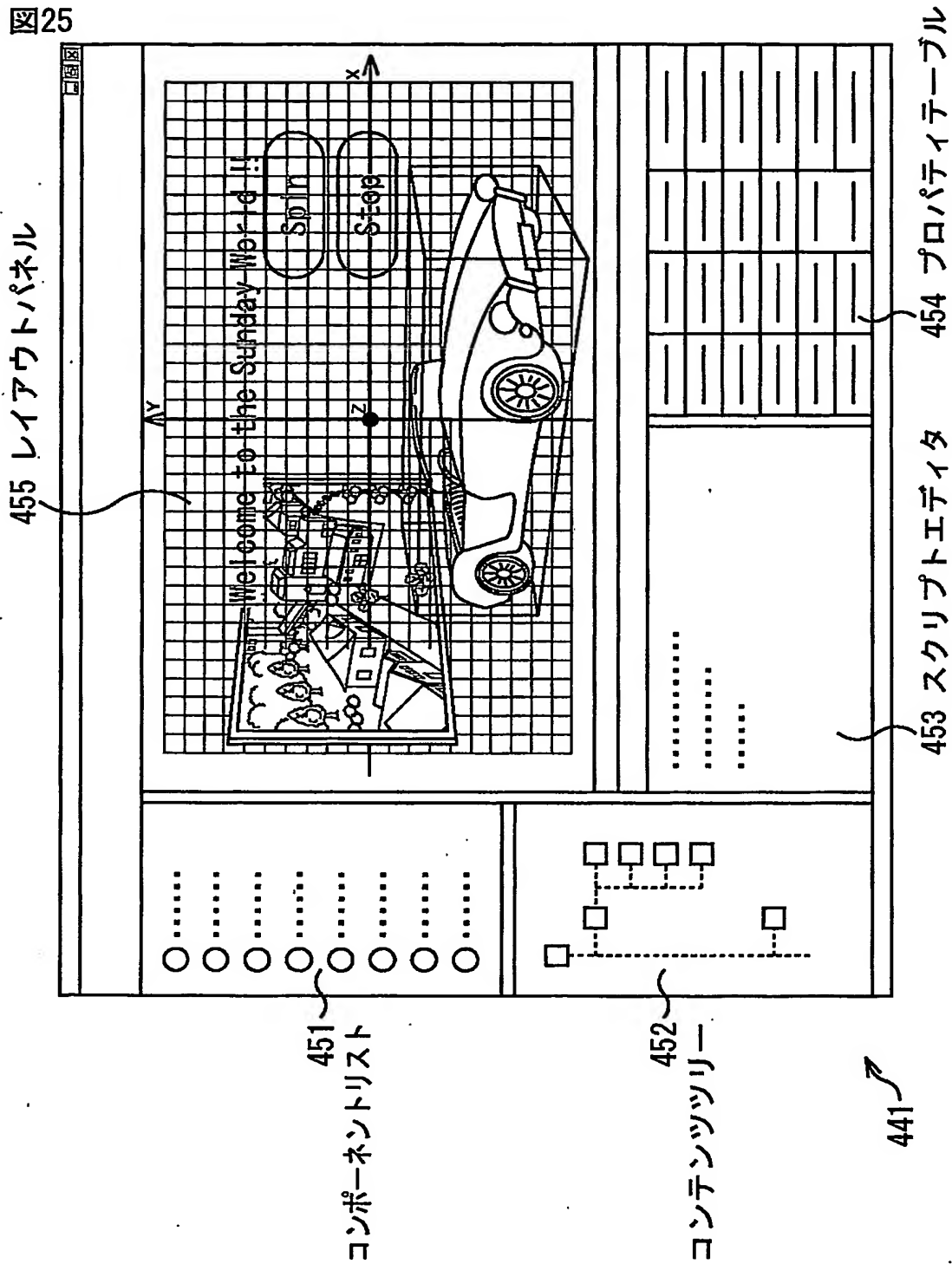
【図 24】

図24



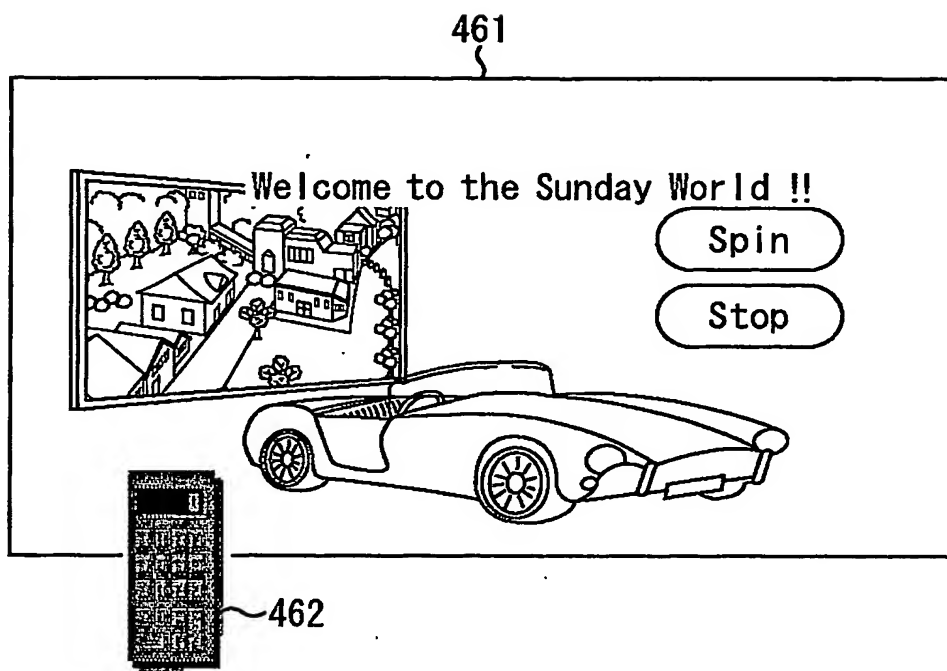
【図 25】

図25



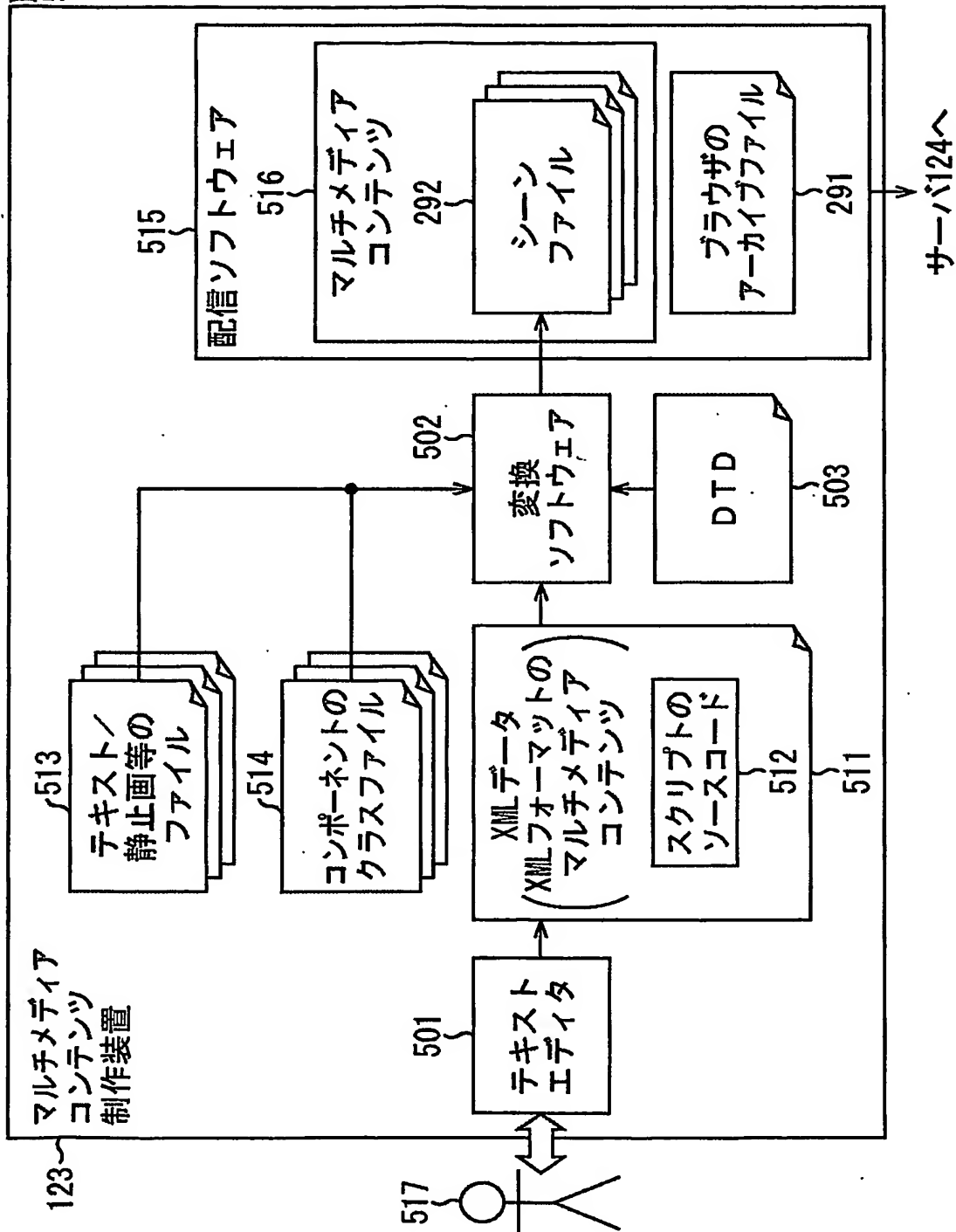
【図 26】

図26



【図 27】

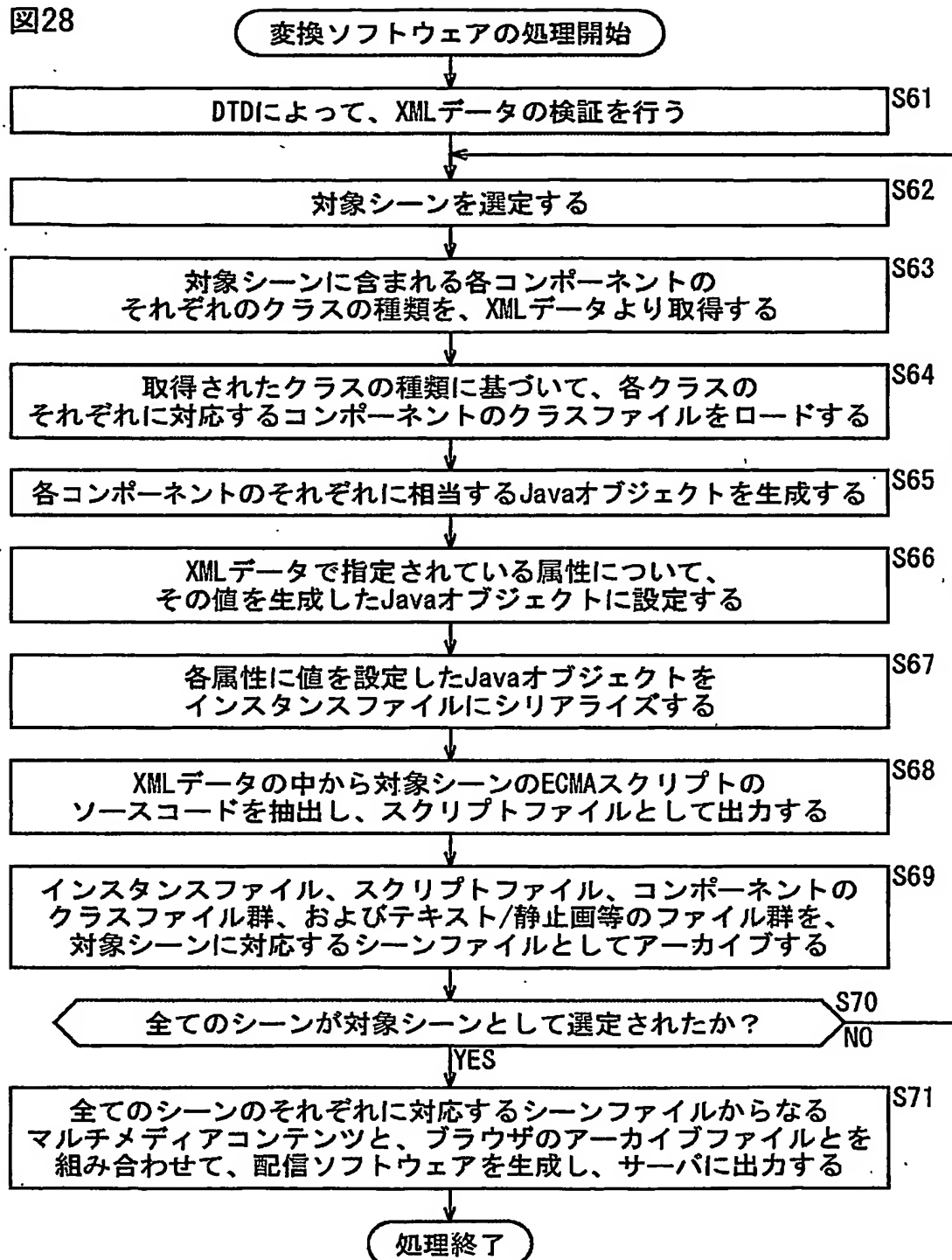
図27





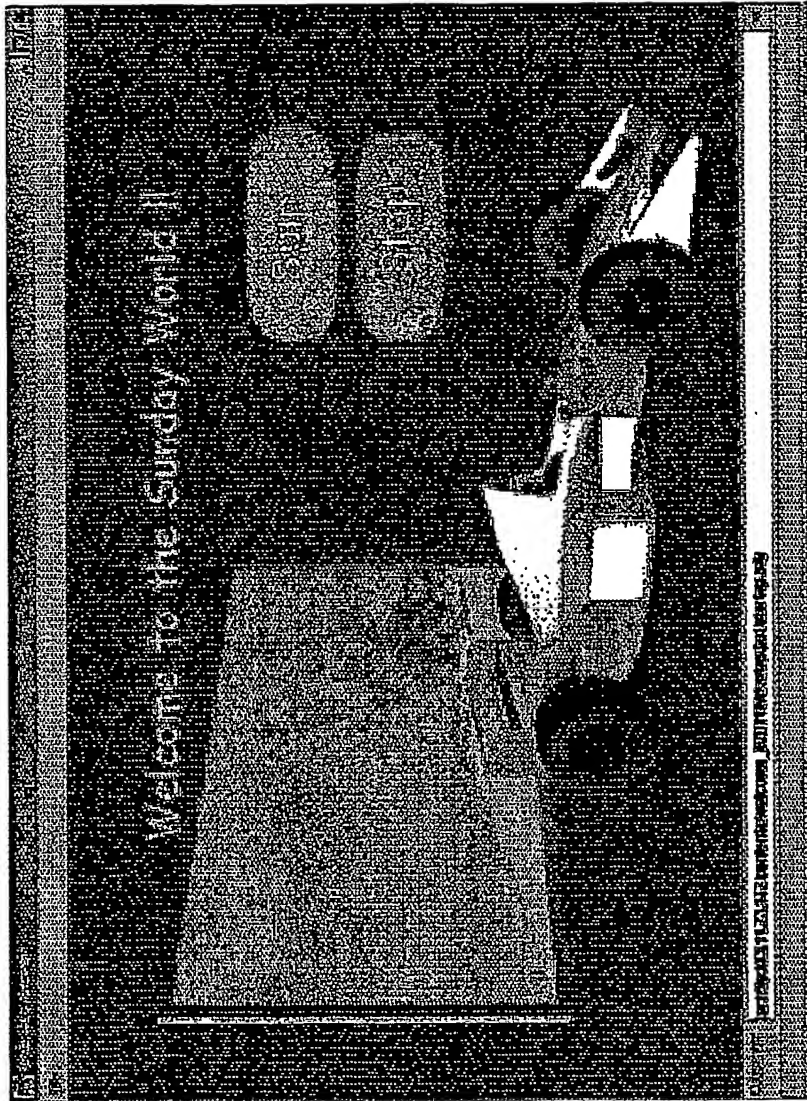
【図 28】

図28



【図29】

図29



521

【図 3 0】

図30

511-1

```

<?xml version="1.0" encoding="UTF-8"?>
<content name="welcome"> ... a
  <scene name="startup"> ... b
    <script>
      <![CDATA[
        // This is ECMAScript Code

        button1.focus();

        function spin() {
          porsche.spin();
        }

        function stopSpin() {
          porsche.stopSpin();
        }

      ]]>
    </script>
    <component class="com.sony.slight.SSpotLight"> ... c
      <property name="id"> ... d
        <string>sSpotLight1</string> ... e
      </property>
      <property name="position">
        <object class="javax.vecmath.Vector3d">
          <constructor>
            <arg name="x">
              <double>1.1</double>
            </arg>
            <arg name="y">
              <double>0.0</double>
            </arg>
            <arg name="z">
              <double>0.5</double>
            </arg>
          </constructor>

```

【図 3 1】

図31

511-2

```
</object>
</property>
<property name="color">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>1.0</float>
      </arg>
      <arg name="y">
        <float>0.7058824</float>
      </arg>
      <arg name="z">
        <float>0.7058824</float>
      </arg>
    </constructor>
  </object>
</property>
<property name="shared">
  <boolean>false</boolean>
</property>
<property name="active">
  <boolean>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="direction">
  <object class="javax.vecmath.Vector3f">
    <constructor>
      <arg name="x">
        <float>-1.0</float>
      </arg>
      <arg name="y">
        <float>-0.6</float>
      </arg>
      <arg name="z">
```

【図 3 2】

図32

511-3

```

        <float>-0.7</float>
      </arg>
    </constructor>
  </object>
</property>
<property name="attenuation">
  <object class="javax.vecmath.Point3f">
    <constructor>
      <arg name="x">
        <float>1.0</float>
      </arg>
      <arg name="y">
        <float>0.5</float>
      </arg>
      <arg name="z">
        <float>0.0</float>
      </arg>
    </constructor>
  </object>
</property>
<property name="concentration">
  <float>0.8</float>
</property>
<property name="spreadAngle">
  <float>0.6</float>
</property>
</component>
<component class="com.sony.dlight.SDirectionalLight">
  <property name="id">
    <string>sDirectionalLight1</string>
  </property>
  <property name="color">
    <object class="javax.vecmath.Color3f">
      <constructor>
        <arg name="x">
          <float>1.0</float>

```

【図 3 3】

図33

511-4

```

    </arg>
    <arg name="y">
      <float>1.0</float>
    </arg>
    <arg name="z">
      <float>1.0</float>
    </arg>
  </constructor>
</object>
</property>
<property name="shared">
  <boolean>false</boolean>
</property>
<property name="active">
  <boolean>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="direction">
  <object class="javax.vecmath.Vector3f">
    <constructor>
      <arg name="x">
        <float>1.0</float>
      </arg>
      <arg name="y">
        <float>0.0</float>
      </arg>
      <arg name="z">
        <float>-1.0</float>
      </arg>
    </constructor>
  </object>
</property>
</component>
<component class="com.sony.picture.SPicture">

```

【図 3 4】

図34

511-5

```
<property name="id">
  <string>picture1</string>
</property>
<property name="position">
  <object class="javax.vecmath.Vector3d">
    <constructor>
      <arg name="x">
        <double>-0.8</double>
      </arg>
      <arg name="y">
        <double>0.12000000000000001</double>
      </arg>
      <arg name="z">
        <double>-0.3</double>
      </arg>
    </constructor>
  </object>
</property>
<property name="width">
  <double>1.0</double>
</property>
<property name="height">
  <double>0.56</double>
</property>
<property name="scale">
  <double>2.2</double>
</property>
<property name="shared">
  <boolean>>false</boolean>
</property>
<property name="active">
  <boolean>>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
```

【図 3 5】

図35

511-6

```

<property name="filepath">
  <object class="com. sony. sunday. util. SResourcePath">
    <constructor>
      <arg name="path">
        <string>/startup/background1. JPG</string>
      </arg>
    </constructor>
  </object>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax. vecmath. AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0. 0</double>
      </arg>
      <arg name="y">
        <double>0. 9913287246227138</double>
      </arg>
      <arg name="z">
        <double>0. 0</double>
      </arg>
      <arg name="angle">
        <double>0. 85</double>
      </arg>
    </constructor>
  </object>
</property>
<property name="frameTextureFilepath">
  <object class="com. sony. sunday. util. SResourcePath">
    <constructor>
      <arg name="path">
        <string></string>
      </arg>

```



【図 3 6】

図36

511-7

```

    </constructor>
  </object>
</property>
<property name="frameTextureRepeatTime">
  <double>5.0</double>
</property>
<property name="frameMargin">
  <double>0.04</double>
</property>
<property name="frameTransparency">
  <float>0.0</float>
</property>
<property name="frameMaterial">
  <object class="com.sony.sunday.util.SMaterial">
    <constructor>
      <arg name="ambientColor">
        <object class="javax.vecmath.Color3f">
          <constructor>
            <arg name="x">
              <float>0.0</float>
            </arg>
            <arg name="y">
              <float>0.2</float>
            </arg>
            <arg name="z">
              <float>0.2</float>
            </arg>
          </constructor>
        </object>
      </arg>
      <arg name="emissiveColor">
        <object class="javax.vecmath.Color3f">
          <constructor>
            <arg name="x">
              <float>0.0</float>
            </arg>

```

【図 3 7】

図37

511-8

```
<arg name="y">
  <float>0.0</float>
</arg>
<arg name="z">
  <float>0.0</float>
</arg>
</constructor>
</object>
</arg>
<arg name="diffuseColor">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>0.0</float>
      </arg>
      <arg name="y">
        <float>0.8</float>
      </arg>
      <arg name="z">
        <float>0.8</float>
      </arg>
    </constructor>
  </object>
</arg>
<arg name="specularColor">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>0.0</float>
      </arg>
      <arg name="y">
        <float>0.8</float>
      </arg>
      <arg name="z">
        <float>0.8</float>
      </arg>
    </constructor>
  </object>
</arg>
```

【図 3 8】

図38

511-9

```

        </constructor>
    </object>
</arg>
<arg name="shininess">
    <float>64.0</float>
</arg>
</constructor>
</object>
</property>
<property name="depth">
    <double>0.02</double>
</property>
<property name="showFrame">
    <boolean>true</boolean>
</property>
</component>
<component class="com.sony.pushbutton.SPushButton">
    <property name="id">
        <string>button1</string>
    </property>
    <property name="position">
        <object class="javax.vecmath.Vector3d">
            <constructor>
                <arg name="x">
                    <double>1.2</double>
                </arg>
                <arg name="y">
                    <double>0.38</double>
                </arg>
                <arg name="z">
                    <double>-0.2</double>
                </arg>
            </constructor>
        </object>
    </property>
    <property name="scale">

```

【図 39】

図 39

511-10

```
<double>0.8</double>
</property>
<property name="shared">
  <boolean>>false</boolean>
</property>
<property name="active">
  <boolean>>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="filepath">
  <object class="com.sony.sunday.util.SResourcePath">
    <constructor>
      <arg name="path">
        <string>/startup/buttonNormal.sob</string>
      </arg>
    </constructor>
  </object>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.0</double>
      </arg>
      <arg name="y">
        <double>1.0</double>
      </arg>
      <arg name="z">
        <double>0.0</double>
      </arg>
      <arg name="angle">
```

【図 4 0】

図40

511-11

```

        <double>0.0</double>
      </arg>
    </constructor>
  </object>
</property>
<property name="onfocus">
  <string></string>
</property>
<property name="onblur">
  <string></string>
</property>
<property name="onkeydown">
  <string></string>
</property>
<property name="onkeyup">
  <string></string>
</property>
<property name="navup">
  <string></string>
</property>
<property name="navdown">
  <string>button2</string>
</property>
<property name="navleft">
  <string></string>
</property>
<property name="navright">
  <string></string>
</property>
<property name="focusMaterial">
  <object class="com.sony.sunday.util.SMaterial">
    <constructor>
      <arg name="ambientColor">
        <object class="javax.vecmath.Color3f">
          <constructor>
            <arg name="x">

```

【図 4 1】

図41

511-12

```
<float>0.3</float>
</arg>
<arg name="y">
  <float>0.3</float>
</arg>
<arg name="z">
  <float>0.0</float>
</arg>
</constructor>
</object>
</arg>
<arg name="emissiveColor">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>0.1</float>
      </arg>
      <arg name="y">
        <float>0.1</float>
      </arg>
      <arg name="z">
        <float>0.0</float>
      </arg>
    </constructor>
  </object>
</arg>
<arg name="diffuseColor">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>0.8</float>
      </arg>
      <arg name="y">
        <float>0.8</float>
      </arg>
      <arg name="z">
```

【図 4 2】

図42

511-13

```

        <float>0.0</float>
    </arg>
</constructor>
</object>
</arg>
<arg name="specularColor">
    <object class="javax.vecmath.Color3f">
        <constructor>
            <arg name="x">
                <float>1.0</float>
            </arg>
            <arg name="y">
                <float>1.0</float>
            </arg>
            <arg name="z">
                <float>0.0</float>
            </arg>
        </constructor>
    </object>
</arg>
<arg name="shininess">
    <float>128.0</float>
</arg>
</constructor>
</object>
</property>
<property name="normalGeometryLabel">
    <string>Default</string>
</property>
<property name="selectedGeometryLabel">
    <string>Default_Selected</string>
</property>
<property name="onpressed">
    <string>spin();</string>
</property>
</component>

```

【図 4 3】

図43

511-14

```

<component class="com.sony.pushbutton.SPushButton">
  <property name="id">
    <string>button2</string>
  </property>
  <property name="position">
    <object class="javax.vecmath.Vector3d">
      <constructor>
        <arg name="x">
          <double>1.2</double>
        </arg>
        <arg name="y">
          <double>0.0</double>
        </arg>
        <arg name="z">
          <double>-0.2</double>
        </arg>
      </constructor>
    </object>
  </property>
  <property name="scale">
    <double>0.8</double>
  </property>
  <property name="shared">
    <boolean>false</boolean>
  </property>
  <property name="active">
    <boolean>false</boolean>
  </property>
  <property name="initiallyActive">
    <boolean>true</boolean>
  </property>
  <property name="filepath">
    <object class="com.sony.sunday.util.SResourcePath">
      <constructor>
        <arg name="path">
          <string>/startup/buttonNormal.sob</string>
        </arg>
      </constructor>
    </object>
  </property>
</component>

```



【図 4 4】

図44

511-15

```
</arg>
</constructor>
</object>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.0</double>
      </arg>
      <arg name="y">
        <double>1.0</double>
      </arg>
      <arg name="z">
        <double>0.0</double>
      </arg>
      <arg name="angle">
        <double>0.0</double>
      </arg>
    </constructor>
  </object>
</property>
<property name="onfocus">
  <string></string>
</property>
<property name="onblur">
  <string></string>
</property>
<property name="onkeydown">
  <string></string>
</property>
<property name="onkeyup">
  <string></string>
```

【図 4 5】

図45

511-16

```
</property>
<property name="navup">
  <string>button1</string>
</property>
<property name="navdown">
  <string></string>
</property>
<property name="navleft">
  <string></string>
</property>
<property name="navright">
  <string></string>
</property>
<property name="focusMaterial">
  <object class="com.sony.sunday.util.SMaterial">
    <constructor>
      <arg name="ambientColor">
        <object class="javax.vecmath.Color3f">
          <constructor>
            <arg name="x">
              <float>0.3</float>
            </arg>
            <arg name="y">
              <float>0.3</float>
            </arg>
            <arg name="z">
              <float>0.0</float>
            </arg>
          </constructor>
        </object>
      </arg>
      <arg name="emissiveColor">
        <object class="javax.vecmath.Color3f">
          <constructor>
            <arg name="x">
              <float>0.1</float>
```

【図 4 6】

図46

511-17

```
</arg>
<arg name="y">
  <float>0.1</float>
</arg>
<arg name="z">
  <float>0.0</float>
</arg>
</constructor>
</object>
</arg>
<arg name="diffuseColor">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>0.8</float>
      </arg>
      <arg name="y">
        <float>0.8</float>
      </arg>
      <arg name="z">
        <float>0.0</float>
      </arg>
    </constructor>
  </object>
</arg>
<arg name="specularColor">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>1.0</float>
      </arg>
      <arg name="y">
        <float>1.0</float>
      </arg>
      <arg name="z">
        <float>0.0</float>
      </arg>
    </constructor>
  </object>
</arg>
```

【图 4 7】

图47

511-18

```

        </arg>
      </constructor>
    </object>
  </arg>
  <arg name="shininess">
    <float>128.0</float>
  </arg>
</constructor>
</object>
</property>
<property name="normalGeometryLabel">
  <string>Default</string>
</property>
<property name="selectedGeometryLabel">
  <string>Default_Selected</string>
</property>
<property name="onpressed">
  <string>stopSpin();</string>
</property>
</component>
<component class="com.sony.label.SLabel">
  <property name="id">
    <string>label1</string>
  </property>
  <property name="position">
    <object class="javax.vecmath.Vector3d">
      <constructor>
        <arg name="x">
          <double>0.9200000000000002</double>
        </arg>
        <arg name="y">
          <double>0.2799999999999999</double>
        </arg>
        <arg name="z">
          <double>0.3</double>
        </arg>
      </constructor>
    </object>
  </property>

```

【図 4 8】

図48

511-19

```
</constructor>
</object>
</property>
<property name="font">
  <object class="java.awt.Font">
    <constructor>
      <arg name="name">
        <string>Arial</string>
      </arg>
      <arg name="style">
        <int>1</int>
      </arg>
      <arg name="size">
        <int>36</int>
      </arg>
    </constructor>
  </object>
</property>
<property name="color">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>1.0</float>
      </arg>
      <arg name="y">
        <float>1.0</float>
      </arg>
      <arg name="z">
        <float>1.0</float>
      </arg>
    </constructor>
  </object>
</property>
<property name="text">
  <string>Spin</string>
</property>
```

【図 4 9】

図49

511-20

```
<property name="scale">
  <double>1.0</double>
</property>
<property name="shared">
  <boolean>>false</boolean>
</property>
<property name="active">
  <boolean>>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.0</double>
      </arg>
      <arg name="y">
        <double>1.0</double>
      </arg>
      <arg name="z">
        <double>0.0</double>
      </arg>
      <arg name="angle">
        <double>0.0</double>
      </arg>
    </constructor>
  </object>
</property>
</component>
<component class="com.sony.label.SLabel">
  <property name="id">
```

【図 5 0】

図50

511-21

```
<string>label2</string>
</property>
<property name="position">
  <object class="javax.vecmath.Vector3d">
    <constructor>
      <arg name="x">
        <double>0.94</double>
      </arg>
      <arg name="y">
        <double>-0.05999999999999993</double>
      </arg>
      <arg name="z">
        <double>0.3</double>
      </arg>
    </constructor>
  </object>
</property>
<property name="font">
  <object class="java.awt.Font">
    <constructor>
      <arg name="name">
        <string>Arial</string>
      </arg>
      <arg name="style">
        <int>1</int>
      </arg>
      <arg name="size">
        <int>36</int>
      </arg>
    </constructor>
  </object>
</property>
<property name="color">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
```

【図 5 1】

図51

511-22

```

    <float>1.0</float>
  </arg>
  <arg name="y">
    <float>1.0</float>
  </arg>
  <arg name="z">
    <float>1.0</float>
  </arg>
</constructor>
</object>
</property>
<property name="text">
  <string>Stop</string>
</property>
<property name="scale">
  <double>1.0</double>
</property>
<property name="shared">
  <boolean>>false</boolean>
</property>
<property name="active">
  <boolean>>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.0</double>
      </arg>
      <arg name="y">

```



【図 5 2】

図52

511-23

```

    <float>1.0</float>
  </arg>
  <arg name="y">
    <float>1.0</float>
  </arg>
  <arg name="z">
    <float>1.0</float>
  </arg>
</constructor>
</object>
</property>
<property name="text">
  <string>Stop</string>
</property>
<property name="scale">
  <double>1.0</double>
</property>
<property name="shared">
  <boolean>>false</boolean>
</property>
<property name="active">
  <boolean>>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.0</double>
      </arg>
      <arg name="y">

```

【図 5 3】

図 53

511-24

```

    <double>1.0</double>
  </arg>
  <arg name="z">
    <double>0.0</double>
  </arg>
  <arg name="angle">
    <double>0.0</double>
  </arg>
</constructor>
</object>
</property>
</component>
<component class="com.sony.label.SLabel">
  <property name="id">
    <string>title</string>
  </property>
  <property name="position">
    <object class="javax.vecmath.Vector3d">
      <constructor>
        <arg name="x">
          <double>-1.0000000000000004</double>
        </arg>
        <arg name="y">
          <double>0.6600000000000001</double>
        </arg>
        <arg name="z">
          <double>0.3</double>
        </arg>
      </constructor>
    </object>
  </property>
  <property name="font">
    <object class="java.awt.Font">
      <constructor>
        <arg name="name">
          <string>Comic Sans MS</string>

```

【図 5 4】

図54

511-25

```

</arg>
<arg name="style">
  <int>1</int>
</arg>
<arg name="size">
  <int>36</int>
</arg>
</constructor>
</object>
</property>
<property name="color">
  <object class="javax.vecmath.Color3f">
    <constructor>
      <arg name="x">
        <float>0.6</float>
      </arg>
      <arg name="y">
        <float>0.8</float>
      </arg>
      <arg name="z">
        <float>0.0</float>
      </arg>
    </constructor>
  </object>
</property>
<property name="text">
  <string>Welcome to the Sunday World !!</string>
</property>
<property name="scale">
  <double>1.0</double>
</property>
<property name="shared">
  <boolean>false</boolean>
</property>
<property name="active">
  <boolean>false</boolean>

```

【図 5 5】

図55

511-26

```

</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.0</double>
      </arg>
      <arg name="y">
        <double>1.0</double>
      </arg>
      <arg name="z">
        <double>0.0</double>
      </arg>
      <arg name="angle">
        <double>0.0</double>
      </arg>
    </constructor>
  </object>
</property>
</component>
<component class="com.sony.shape.S3DShape">
  <property name="id">
    <string>porsche</string>
  </property>
  <property name="position">
    <object class="javax.vecmath.Vector3d">
      <constructor>
        <arg name="x">
          <double>1.3400000000000003</double>
        </arg>

```

【図 5 6】

図56

511-27

```

    <arg name="y">
      <double>-0.7599999999999998</double>
    </arg>
    <arg name="z">
      <double>0.7600000000000001</double>
    </arg>
  </constructor>
</object>
</property>
<property name="width">
  <double>1.0</double>
</property>
<property name="height">
  <double>1.0</double>
</property>
<property name="scale">
  <double>0.55</double>
</property>
<property name="shared">
  <boolean>false</boolean>
</property>
<property name="active">
  <boolean>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="filepath">
  <object class="com.sony.sunday.util.SResourcePath">
    <constructor>
      <arg name="path">
        <string>/startup/formula0ne.sob</string>
      </arg>
    </constructor>
  </object>
</property>

```

【図 5 7】

図57

511-28

```

<property name="renderingOrder">
  <int>0</int>
</property>
<property name="rotation">
  <object class="javax.vecmath.AxisAngle4d">
    <constructor>
      <arg name="x">
        <double>0.023045814612553756</double>
      </arg>
      <arg name="y">
        <double>-0.9993246794860199</double>
      </arg>
      <arg name="z">
        <double>-0.028619493339460032</double>
      </arg>
      <arg name="angle">
        <double>1.9958596229553223</double>
      </arg>
    </constructor>
  </object>
</property>
<property name="depth">
  <double>1.0</double>
</property>
</component>
<component class="com.sony.colorbg.SColorBackground">
  <property name="id">
    <string>sColorBackground1</string>
  </property>
  <property name="color">
    <object class="javax.vecmath.Color3f">
      <constructor>
        <arg name="x">
          <float>0.10980392</float>
        </arg>
        <arg name="y">

```

【図 5 8】

図58

511-29

```

    <float>0.11764706</float>
  </arg>
  <arg name="z">
    <float>0.11764706</float>
  </arg>
</constructor>
</object>
</property>
<property name="shared">
  <boolean>>false</boolean>
</property>
<property name="active">
  <boolean>>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>>true</boolean>
</property>
</component>
<component class="com.sony.alight.SAmbientLight">
  <property name="id">
    <string>aLight</string>
  </property>
  <property name="color">
    <object class="javax.vecmath.Color3f">
      <constructor>
        <arg name="x">
          <float>1.0</float>
        </arg>
        <arg name="y">
          <float>1.0</float>
        </arg>
        <arg name="z">
          <float>1.0</float>
        </arg>
      </constructor>
    </object>
  </property>

```

【図 5 9】

図59

511-30

```
</property>
<property name="shared">
  <boolean>false</boolean>
</property>
<property name="active">
  <boolean>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
</component>
<component class="com.sony.slight.SSpotLight">
  <property name="id">
    <string>sSpotLight1_1</string>
  </property>
  <property name="position">
    <object class="javax.vecmath.Vector3d">
      <constructor>
        <arg name="x">
          <double>-0.54000000000000006</double>
        </arg>
        <arg name="y">
          <double>-0.039999999999999998</double>
        </arg>
        <arg name="z">
          <double>0.7</double>
        </arg>
      </constructor>
    </object>
  </property>
  <property name="color">
    <object class="javax.vecmath.Color3f">
      <constructor>
        <arg name="x">
          <float>0.7019608</float>
        </arg>
```



【図 6 0】

図60

511-31

```
<arg name="y">
  <float>1.0</float>
</arg>
<arg name="z">
  <float>1.0</float>
</arg>
</constructor>
</object>
</property>
<property name="shared">
  <boolean>false</boolean>
</property>
<property name="active">
  <boolean>false</boolean>
</property>
<property name="initiallyActive">
  <boolean>true</boolean>
</property>
<property name="direction">
  <object class="javax.vecmath.Vector3f">
    <constructor>
      <arg name="x">
        <float>1.0</float>
      </arg>
      <arg name="y">
        <float>-0.6</float>
      </arg>
      <arg name="z">
        <float>-0.7</float>
      </arg>
    </constructor>
  </object>
</property>
<property name="attenuation">
  <object class="javax.vecmath.Point3f">
    <constructor>
```

【図 6 1】

図61

511-32

```
<arg name="x">
  <float>1.0</float>
</arg>
<arg name="y">
  <float>0.5</float>
</arg>
<arg name="z">
  <float>0.0</float>
</arg>
</constructor>
</object>
</property>
<property name="concentration">
  <float>0.8</float>
</property>
<property name="spreadAngle">
  <float>0.6</float>
</property>
</component>
</scene>
</content>
```

【図 6 2】

図62

503-1

<!ELEMENT content (scene+)> テンツ (最上位要素) →	<!-- コン
<!ATTLIST content name CDATA #IMPLIED> テンツの名称 (任意指定) →	<!-- コン
<!ELEMENT scene (script, component*)> テンツを構成するシーン →	<!-- コン
<!ATTLIST scene name ID #REQUIRED> ンの名称 (必須指定) →	<!-- シー
<!ELEMENT script (#PCDATA)> スクリプトコード →	<!-- ECMA
<!ELEMENT component (property*)> ンを構成するコンポーネント →	<!-- シー
<!ATTLIST component class CDATA #REQUIRED> ポーネントのクラス名 (必須指定) →	<!-- コン
<!ELEMENT property (boolean   byte   char   double   float   int   long   short   string   object)> ポーネントのプロパティ値 →	<!-- コン
<!ATTLIST property name CDATA #REQUIRED> パティ (フィールド) 名 (必須指定) →	<!-- プロ
<!ELEMENT boolean (#PCDATA)> ean (プリミティブ) 値 →	<!-- bool
<!ELEMENT byte (#PCDATA)> (プリミティブ) 値 →	<!-- byte
<!ELEMENT char (#PCDATA)> (プリミティブ) 値 →	<!-- char
<!ELEMENT double (#PCDATA)> le (プリミティブ) 値 →	<!-- doub
<!ELEMENT float (#PCDATA)> t (プリミティブ) 値 →	<!-- floa
<!ELEMENT int (#PCDATA)> (プリミティブ) 値 →	<!-- int
<!ELEMENT long (#PCDATA)> (プリミティブ) 値 →	<!-- long

【図 6 3】

図63

503-2

```

<!ELEMENT short (#PCDATA)>                                <!-- shor
t (プリミティブ) 値 -->
<!ELEMENT string (#PCDATA)>                                <!-- stri
ng (プリミティブ) 値 -->
<!ELEMENT object ((constructor?, setter*) | reference)>
  <!-- オブジェクト -->
<!ATTLIST object class CDATA #REQUIRED                    <!-- clas
                      name CDATA #IMPLIED>                <!--
s:クラス名 (必須指定) name:参照名 (任意指定) -->
<!ELEMENT constructor (arg+)>                              <!-- コン
ストラクタ、セッターメソッドに渡す引数 -->
<!ELEMENT setter (arg)>                                     <!-- セッ
ターメソッド -->
<!ATTLIST setter name CDATA #REQUIRED>                    <!-- メソ
ッド名 -->
<!ELEMENT arg (boolean | byte | char | double
               | float | int | long | short
               | string | object)>                          <!-- 引数
-->
<!ATTLIST arg name CDATA #IMPLIED>                          <!-- 引数
名 (任意指定) -->
<!ELEMENT reference (#PCDATA)>                             <!-- 他オ
ブジェクトへの参照 (環状リスト構造の表現用) -->

<!-- 実際にこのファイルを用いて検証を行う場合は、事前にこ
のファイルの文字エンコードをUTF-8に変換する必要がある -->

```

【書類名】 要約書

【要約】 高機能なマルチメディアコンテンツを容易に取り扱うことができるようにする。

【解決手段】 受信端末 1 1 3 は、ネットワーク 1 1 2 上の、入力された URL に対応するサーバ 1 2 4 にアクセスする。サーバ 1 2 4 は、所定のサービスをコンポーネントが受けるために満たすべき仕様を有するブラウザソフトウェア 2 5 1 と、所定の構成を有するマルチメディアコンテンツ 2 5 2 とからなる配信ソフトウェア 2 4 3 を、ネットワーク 1 1 2 を介して受信端末 1 1 3 に配信する。受信端末 1 1 3 は、配信されたブラウザ 2 5 1 上に、配信されたマルチメディアコンテンツ 2 5 2 を表示させる。本発明は、パーソナルコンピュータ、デジタル家電製品等に適用できる。

【選択図】 図 1 3

認定・付加情報

特許出願の番号	特願 2002-180926
受付番号	50200904252
書類名	特許願
担当官	第八担当上席 0097
作成日	平成14年 6月26日

<認定情報・付加情報>

【特許出願人】

【識別番号】

000002185

【住所又は居所】

東京都品川区北品川6丁目7番35号

【氏名又は名称】

ソニー株式会社

【代理人】

申請人

【識別番号】

100082131

【住所又は居所】

東京都新宿区西新宿7丁目11番18号 711

ビルディング4階 稲本国際特許事務所

【氏名又は名称】

稲本 義雄

出 願 人 履 歴 情 報

識別番号

[000002185]

1. 変更年月日

1990年 8月30日

[変更理由]

新規登録

住 所

東京都品川区北品川6丁目7番35号

氏 名

ソニー株式会社

**This Page is Inserted by IFW Indexing and Scanning  
Operations and is not part of the Official Record**

**BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☒ FADED TEXT OR DRAWING
- ☐ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☒ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: \_\_\_\_\_

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.**